

Runtime-optimised Intra Coding Mode Selection for Real-time H.264/AVC Video Coding

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Magister

im Rahmen des Studiums

Informatikmanagement

eingereicht von

Dipl.-Ing. Dr. techn. Harald Jordan

Matrikelnummer 9825829

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung

Betreuer/in: A.o. Univ.-Prof. Mag. Dipl.-Ing. Dr. techn. Margrit Gelautz

Mitwirkung: Dipl.-Ing. Florian Seitner, Dipl.-Ing. Dr. techn. Michael Bleyer

Wien, 03.01.2011

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Harald Jordan
Wimbergergasse 21/12
1070 Wien

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Wien, 03.01.2011

Unterschrift

Abstract

H.264 is a powerful video compression standard using advanced spatial and temporal encoding techniques. Compared to the MPEG4 Part 2 standard, H.264 achieves the same visual quality at half of the bitrate.

The coding efficiency of H.264 makes it well suitable for quality- and time-critical video applications such as low-latency video transmission. This work concentrates on a part of the H.264 coding process that is used in many of these applications — the spatial (Intra) prediction. This coding tool propagates pixels in a frame for predicting unknown neighbouring pixel regions. H.264 supports multiple propagation patterns which are referred to as *coding modes*. The challenge of most real-time video encoders is to choose an efficient coding mode in a reasonable amount of time. This process is called coding mode selection (CMS) and involves the computation of all possible coding modes and choosing the best one.

In this thesis, we focus on reducing the computational complexity of the CMS. First, we investigate the behaviour of the CMS on a set of test sequences. The impact on the coding efficiency when using only a reduced number of coding modes is analyzed. Second, we develop two methods for speeding up the CMS. Both methods skip the computation of coding modes that are unlikely to improve the coding efficiency. This results in a significant reduction in the computational complexity of the encoder.

For each of the two proposed CMS algorithms, we evaluate the impact on the coding efficiency, the runtime reduction and the quality. We demonstrate that they reduce the computational complexity of the CMS with no significant degradation of the bitrate and quality.

Kurzfassung

H.264 ist ein leistungsfähiger Standard zur Videokompression, der erweiterte Methoden zur zeitlichen und räumlichen Kodierung benutzt. Im Vergleich zu früheren Standards erreichen H.264 konforme Encoder höhere Kompressionsraten bei gleichbleibender Bildqualität.

Die vorliegende Arbeit beschäftigt sich mit einem Teilbereich von H.264, der speziell für qualitäts- und zeitkritische Anwendungen von Bedeutung ist. Es handelt sich hierbei um die räumliche (Intra) Kodierung. Zur Vorausberechnung von Bildpunkten innerhalb eines Bildes benutzt dieses Verfahren die vorhandenen Bildpunkte aus der direkten Nachbarschaft und einen Satz von speziellen Kodierungsmustern. Diese Kodierungsmuster werden üblicherweise als *coding modes* bezeichnet. Eine große Herausforderung von echtzeitfähigen Encodern ist die effiziente und zeitgerechte Wahl des geeigneten Kodierungsmusters. Der Fachbegriff für diesen Vorgang lautet *coding mode selection (CMS)* und beschreibt die Wahl des besten Kodierungsmusters durch Anwenden aller vorhandenen Muster und dem Vergleich der Ergebniswerte.

Das Ziel dieser Arbeit ist das Herausarbeiten und Entwickeln von Möglichkeiten zur Verringerung des Berechnungsaufwandes dieser CMS. Unter Zuhilfenahme von 12 Testsequenzen wird dazu in einem ersten Schritt das grundsätzliche Betriebsverhalten der CMS untersucht. Der zweite Schritt befasst sich mit der Entwicklung von zwei geeigneten Methoden zur Effizienzsteigerung der CMS. Beide Methoden eliminieren Kodierungsmuster, die mit hoher Wahrscheinlichkeit keine Verbesserung des Kodierungsergebnisses bringen. Diese Vorgehensweise resultiert in einer wesentlichen Herabsetzung des Berechnungsaufwandes und einer Verbesserung des Laufzeitverhaltens.

Die Auswirkungen dieser dynamischen Reduktion von Kodierungsmustern werden für beide Methoden im Anschluss an die Implementierung analysiert. Von besonderem Interesse sind dabei die Änderungen des Laufzeitverhaltens und der Kompressionsqualität. Die Ergebnisse zeigen, dass eine deutliche Reduktion des Berechnungsaufwandes ohne signifikante Verschlechterung der Qualität oder Bitrate erreicht werden kann.

Acknowledgment

“If you have knowledge, let others light their candles at it.”

Margaret Fuller (1810 - 1850)

According to this quotation I would like to express my gratitude to those people who gave me the possibility to conduct this thesis.

Special thanks go to my supervisors *Florian Seitner*, *Michael Bleyer* and *Prof. Margrit Gelautz*.

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Related Work	2
1.2.1	Optimised ICMS Working Directly on the Pixel Data	2
1.2.2	Optimised ICMS Using Coding Information of Neighbouring Blocks	4
1.3	Structure of the Thesis	5
2	H.264 Video Coding	6
2.1	Historical Background	6
2.2	H.264 Encoding and Decoding	6
2.3	Intra Prediction	9
3	Intra Prediction Analysis	19
3.1	Test Environment	19
3.2	Quality Metric	19
3.3	Test Sequences	20
3.4	Motion Analysis of Sequences	21
3.4.1	Metric for the Motion Analysis	21
3.4.2	Motion Analysis	23
3.5	Analysing the Impact of the Block Size	27
3.6	Reduced Number of Intra-Coding Modes	30
3.7	Intra-Only Encoding of Static Regions	33
3.7.1	Mode Retention of Static Regions	34
3.7.2	Temporal Changes of the SAD-Costs	37
3.7.3	Correlation Between Neighbouring Static Sub-Blocks	38
3.7.4	Résumé	41
4	Single Criterion Coding Mode Reuse (SCCMR)	42
4.1	Impact on Runtime Performance	43
4.2	Results	45
4.2.1	Reuse of Coding Modes	45
4.2.2	Coding Efficiency	46
4.3	Conclusion	48
5	Multi Criteria Coding Mode Reuse (MCCMR)	49

5.1	Proposed Algorithm	49
5.1.1	Criterion c_1	50
5.1.2	Criterion c_2 and c_3	50
5.1.3	Criterion c_4	51
5.2	Evaluation of the Modified Encoder	51
5.2.1	Data-Rates	51
5.2.2	Visual Quality	52
5.2.3	Runtime Behaviour	52
5.3	Comparison Between SCCMR and MCCMR	53
5.3.1	Data-Rate Comparison	53
5.3.2	Quality Comparison	54
5.3.3	Improvements of the Runtime Behaviour	55
6	Conclusion	57
	Glossary and Acronyms	58
A	Appendix	60
A.1	Prediction Table	60
	Literature	63
	List of Figures	65
	List of Tables	69

1 Introduction

H.264 represents a powerful and challenging video compression standard with improved compression efficiency and image quality compared to its predecessors such as MPEG-2 and H.262. This is achieved by introducing improved coding tools for removing temporal, spatial and statistical redundancies. However, these tools result in an increased computational complexity that often makes H.264 real-time encoding intractable on many hardware configurations. In this section, we will describe the performance bottleneck most real-time encoders have to deal with (Section 1.1). Various methods for solving this bottleneck have been introduced in previous work. Section 1.2 provides an overview of the existing work.

1.1 Problem Statement

For many modern real-time video applications, encoding solutions based on Intra coding are used. This enables the compression of the video data in an efficient way and keeps the requirements on the hardware in terms of computation power and memory requirements low. However, the H.264 intra coding is based on a computational complex prediction scheme which is challenging for most embedded hardware platforms. For enabling real-time intra coding on these architectures, a compromise between coding efficiency and complexity is required. This is done by using a reduced number of H.264 intra coding modes for the encoding. This lowers the complexity but results in a less efficient prediction process.

A simple strategy would involve the usage of a constant sub-set of coding modes for the coding mode selection (CMS). However, as each coding mode addresses a unique texture pattern, it is most likely that removing some coding modes from the CMS will result in a strongly reduced coding efficiency. Consequently, the CMS requires methods for dynamically selecting good coding mode sub-sets. However, this dynamic selection of the ‘most probable’ coding modes is not straight-forward. The quality of the coding mode prediction has to be estimated based on available information. In the next section, we describe approaches that differ in the way they predict the ‘efficiency’ of a coding mode. This allows us to avoid the computation of coding modes that are unlikely to increase the coding efficiency.

1.2 Related Work

According to Section 1.1, the improvement of the Intra coding mode selection (ICMS) process can be identified as a major starting-point for lowering the encoding complexity. In the recent past, many scientific investigations have been conducted for optimising the H.264 coding mode selection. The following summary provides an overview of related works. The existing approaches can be divided into two major groups. These groups are described in the following sections.

1.2.1 Optimised ICMS Working Directly on the Pixel Data

The basic idea of these methods is to reduce the set of mode candidates. This lowers the computational complexity of the ICMS. The decision on whether to skip a coding mode or not is based on the pattern of the involved pixels and varies with the tools used for analysing these pixels.

Wei et al. [1] proposed an Intra coding mode selection algorithm by using a fast edge detection method. Detecting edges is done by using a non-normalised *Haar*-transformation (NNHT). *Li* and *Ngan* proposed another efficient and fast method for classifying edge blocks by using a normalised *Haar*-transformation (NHT) ([2]).

The method presented in [3] is similar to these approaches. *Yang* et al. proposed an efficient Intra- 4×4 mode decision algorithm, based on the observation of the sub-macroblock properties. These properties are calculated by using Mean Absolute Differences (MAD)-values of horizontal, vertical and plane directions. According to the MAD-results only the most probable coding modes are examined.

Hwang et al. [4] proposed an efficient H.264 Intra-mode decision scheme, using *image structure tensors*. These *image structure tensors* provide local information concerning the structure of a macroblock. This structure information is used to select the most probable coding modes.

Comparing Sum of Absolute Differences (SAD)-values of co-located macroblocks within consecutive frames can be used for early terminating the coding mode selection. This method had been analysed by *Mithun* [5]. Using partially evaluated SADs as abort trigger is described in [6]. These methods are dedicated for Intra-encoded macroblocks within *P-slices*. They are used to decide whether a macroblock is more likely to be Inter-coded or Intra-coded (Fast moving objects or objects entering the scene are more likely to be encoded using Intra-modes. On the other hand static areas tend to use Inter coding modes as these techniques are exploiting their ability to reduce temporal redundancies.).

For efficiently determining the partition-size (16×16 or 4×4) of a macroblock, *Kim* and *Jeong* [7] use texture features of a given macroblock for determining the most probable Intra coding mode.

Another promising approach for deciding the block-size (16×16 or 4×4) is given by *Ryu* and *Kim* [8]. This method analyses the pixel-difference of horizontal and vertical pixel-slices at the borders of a given macroblock. For deciding the optimal block-size, the intensity values of these slices are compared to the values of the previous frame.

A recent approach uses intensity-distributions. It is presented by *Tsai* et al. [9]. The described method introduces a preprocessing stage for extracting orientation features from a macroblock. The algorithm uses a gradient-filter described in [10]. The so evaluated features are used to determine the most probable Intra coding modes.

In [11], *Hwang* et al. proposed a novel H.264 Intra-mode decision scheme for optimising the Intra coding mode selection inside P-slices.

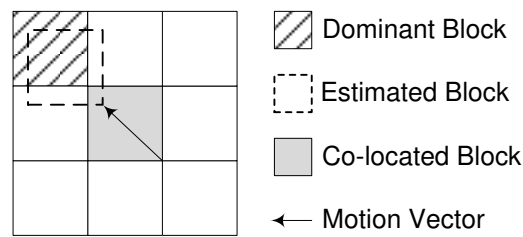


Figure 1.1: Coding Mode selection using temporal correlation. This figure shows the relation of the dominant block, estimated block, co-located block and the motion vector.

The described method thereby uses the temporal correlation of the predicted pixel-block and a corresponding block in a reference frame. The position of the corresponding block is determined by moving the co-located block (see Figure 1.1) towards the location evaluated by the motion-estimation. According to *Hwang*, this method reduces the computational complexity while maintaining similar Peak Signal-to-Noise Ratio (PSNR)-values and data-rates.

Similar approaches are presented in [11], [12] and [13]. *Ko* et al. [12] state that the Intra coding mode of stationary macroblocks will remain as there are no disturbances caused by motion shifts. The reuse scheme thereby references to the co-located position in a given reference-frame.

A fundamentally different method is presented by *Wang* et al. [14]. The authors introduce an interesting method for lowering the complexity of the Intra-mode decision. In a first step, this method locates the position of a pixel-block in the previous frame. In the second step, the coding information at this location is extracted and reused.

Hwang et al. propose a method for deciding the optimal Intra coding mode by estimating the pixel-orientations using the *transformation coefficients*¹ of the residuals ([18]). The evaluation of the pixel-orientation uses *Tsukuba*'s method (see [19]). In this work, *Tsukuba* et al. estimate the pixel-orientation of a macroblock by evaluating the *frequency domain*. Firstly, the current block is transformed by using a Discrete Cosine Transformation (DCT). Secondly, the vertical and horizontal energies are calculated. The resulting energies are determining the best Intra coding mode.

1.2.2 Optimised ICMS Using Coding Information of Neighbouring Blocks

Practical demonstrations showed that the above described methods are suitable for lowering the computational complexity of the ICMS. However, the major drawback is the computational effort of the implemented mode skipping mechanism. To avoid this, the following methods derive the necessary information for the mode skipping from surrounding and already encoded neighbours.

Jafariand Kasaei [20] presented a fast H.264 Intra-mode selection strategy, using macroblock properties. These properties are used to select the most probable subset of candidate modes. This reduction of coding modes considerably improves the encoder's runtime.

In [21], *Fritts* et al. describe a novel Intra-4×4 coding mode selection scheme by reducing the number of candidate modes. To determine the required modes, *Fritts* acts on the assumption that Intra-4×4 coding modes of adjacent blocks are highly correlated.

Similar to *Fritts*' method, is the presented approach of *Elyousfi* et al. [22]. A selection scheme is introduced which is based on the idea that the coding information of smaller blocks is similar to that of the surrounding larger blocks. This assumption is used to eliminate unlike coding modes for smaller blocks.

¹Details regarding the *transformation coefficients* can be found in [15], [16] and [17].

1.3 Structure of the Thesis

Chapter 2 describes the H.264 recommendation of the *ITU-T*. Starting with the historical background, the main principles of the investigated video-compression standard are described.

Chapter 3 presents and discusses the utilised methods and tools, used for analysing the behaviour of the H.264 *Intra-mode* encoding. Additionally, the main characteristics of the H.264 Intra-mode encoding are analysed. This analysis explores the influences of spatial motion and the restriction of coding modes under varying conditions.

In Chapter 4, a first approach for lowering the encoding complexity of Intra-only encoding is presented. Based on the conclusions of Chapter 2, the proposed method reuses the encoding information of static 4×4 -blocks.

An improved version of this algorithm is presented in Chapter 5. The major goal of this novel method is to improve the detection of static 4×4 -blocks, as this lowers the drawbacks of the simplified ICMS.

The most important conclusions and a summary of this thesis are finally given in Chapter 6.

2 H.264 Video Coding

2.1 Historical Background

H.264/Advanced Video Coding (AVC) is a sophisticated video compression standard published by the ITU-T Video Coding Experts Group (VCEG) in 2003. The target of the initial development was the long-term video compression standard *H.26L* which is fundamentally new to its predecessors. In 2001 the ISO Motion Picture Experts Group (MPEG) joined the development as they recognised the immense benefits of this unique compression standard. These two groups formed the Joint Video Team (JVT) which included experts of both MPEG and VCEG. The goal of this joint development was the design of a video compression standard which is capable of achieving the same quality at half the data-rate compared to the MPEG-4 Part 2 standard.

The final result of this joint development is commonly known under two names. The ITU/VCEG specifies the standard as *ITU-T H.264* and the MPEG consortium as MPEG-4/AVC. The reader is referred to [23] for further details.

2.2 H.264 Encoding and Decoding

In analogy to preceding video coding standards such as H.262 and H.263, H.264 specifies the syntax of an H.264 compliant bitstream and how it can be decoded. The encoding process is not explicitly specified. The only requirement on an H.264 encoder is that its output has to be an H.264 compliant bitstream.

In the following sections the structure of H.264 encoding and decoding will be discussed.

Structure of the Encoder

H.264 encoders typically incorporate two data paths, a *forward* path (Figure 2.1 left to right) and the *reconstruction* path (Figure 2.1 right to left). The forward path encodes each frame of the inputstream and delivers the compressed pendant. The reconstruction path decodes previously encoded frames and provides this information for future prediction steps.

In the forward path of Figure 2.1, the current frame F_n is encoded using either spatial (Intra) or temporal (Inter) encoding methods. The temporal prediction uses motion

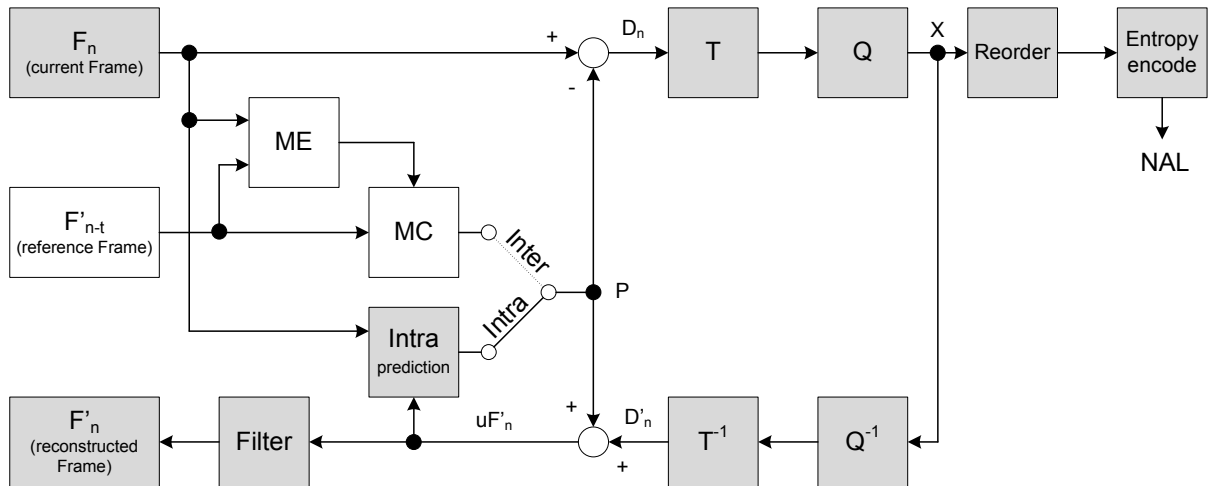


Figure 2.1: Block-diagram of an H.264 encoder. The gray shaded blocks are representing an encoder only using spatial prediction methods. The white blocks are used for encoders supporting temporal prediction too.

information and exploits the pixel similarity of consecutive frames; i.e. a large portion of frame-information is the same information as used in a previous frame at a different spatial position. The temporal encoding process takes advantage of this fact and provides mechanisms for referencing image portions between frames. The spatial encoding only uses the visual information of a single frame making the encoder much easier.

The prediction P is then subtracted from the original pixel-information. The resulting difference D_n is called *residual* information. The next steps involve a transformation of the residual information and a quantisation of the transformed residuals (signal X in Figure 2.1). The quantised signal X is finally re-ordered and entropy-encoded. Combining the entropy-encoded coefficients with additional syntax information (macroblock coding mode, quantisation step-size, motion-vector information, etc.) will lead to the compressed bitstream which is finally passed to the Network Abstraction Layer (NAL).

The reconstruction path has its origin after the quantisation step (X). The reconstruction reverts the quantisation and the block transformation. The differences between the resulting signal D'_n and the signal D_n are caused by the lossy quantisation step. To get the reconstructed frame, the sum of P and D'_n must be calculated. The final filter reduces the effects of blocking artefacts occurring at the borders between macroblock (MB)s.

Structure of the Decoder

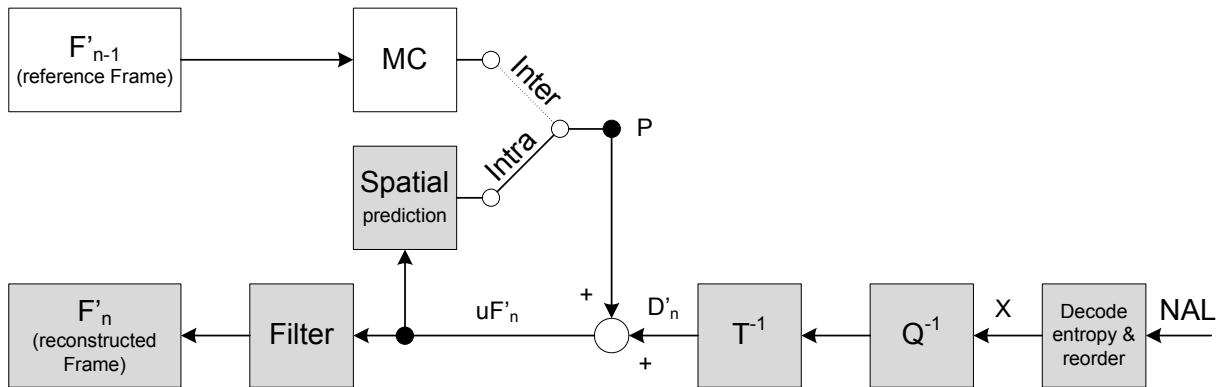


Figure 2.2: Block-diagram of an H.264 decoder. The gray shaded blocks are representing a decoder only using spatial prediction methods. The white blocks are used for decoders supporting temporal decoding.

The input stage of the decoder presented in Figure 2.2 receives the encoded bitstream from the NAL. First the bitstream is entropy decoded (X) and reordered. In the next step, this signal (quantised transform coefficients) is re-scaled by multiplying each coefficient by an integer value for restoring the original scale. These rescaled coefficients are then inverse transformed to get D'_n . For building the prediction P , additional syntax information of the encoded bitstream determines whether to use Intra or Inter prediction. F'_n is finally formed by adding the signals D'_n and P and applying the deblocking filter.

As explained in the last section, H.264 supports two types of prediction, namely the spatial and the temporal prediction. H.264 *Intra* prediction uses information from the current frame whereas H.264 *Inter* prediction uses information from previously encoded frames. In the following sections, both encoding types are discussed whereas the focus is set on the Intra prediction. This is because the thesis targets the runtime-optimisation of H.264 Intra-only encoding.

Profiles

Depending on the application, different encoding functionalities are required. For targeting this issue, H.264 supports three *Profiles*:

- Baseline Profile
- Main Profile

- Extended Profile

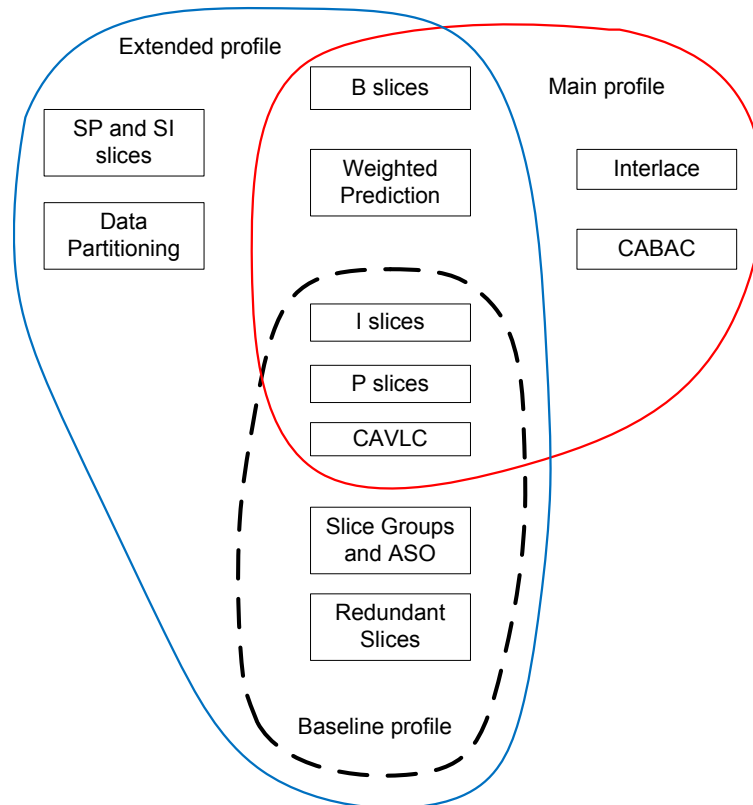


Figure 2.3: H.264 Baseline, Main and Extended profiles.

Each *Profile* defines a particular set of functions (see Figure 2.3). In the presented thesis the functionalities of the *Baseline Profile* are explored as the provided encoder implements this profile only. Further information regarding the other two profiles are given in [24] and [15].

Considering the required complexity of a compliant CODEC (enCODEr / DECODEr), H.264 defines a set of *Levels* whereas each *Level* specifies the limitations in terms of processing rate, frame size and memory requirements.

2.3 Intra Prediction

An H.264 conform video sequence consists of several *frames* (or *fields*), whereas each frame is partitioned into rectangular areas with a constant size of 16×16 pixels. These areas are called macroblocks (MBs).

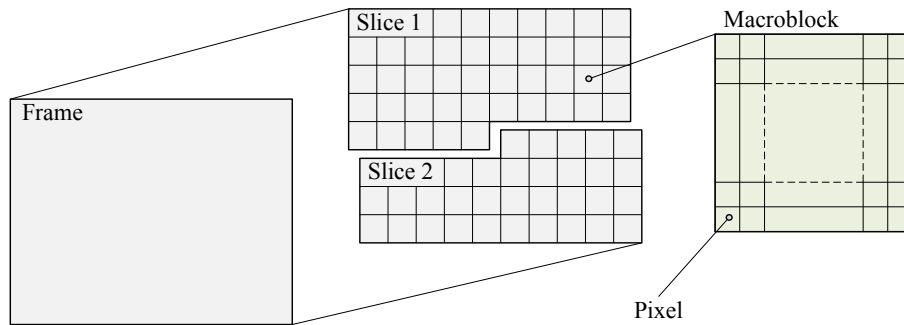


Figure 2.4: Structuring of H.264 into *Frames*, *Slices* and *MBs*: This figure shows a single frame which contains two different Slices (*Slice 1* and *Slice 2*) and several *MBs*.

The encoder processes one MB after the other and produces the bitstream of the compressed frames. Each MB holds luminance-information (luma) as well as chrominance-information (chroma). The dimensions of MBs are different for the luminance and chrominance channels as the resolution differs (see Section 2.3). H.264 specifies that a set of MBs can be grouped to so called *slices*. Figure 2.4 shows that a frame may consist of more than one slice.

The Intra prediction uses pixel information from neighbouring MBs to predict the current MB. H.264 specifies Intra coding modes that describe which pixels are used for predicting the current MB or sub-block.

Intra- 16×16 luma prediction

The MB prediction utilises the luminance samples from adjacent MBs according to the schema given in Figure 2.5. Using the information from the left and upper neighbours for predicting the current MBs pixels is possible because these MBs have already been processed.

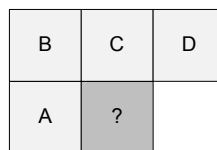


Figure 2.5: Neighbours used for spatial prediction of the current *MB*.

To form the MB prediction from the neighbouring pixel-values, four different coding modes are defined (*Mode 0 - Mode 3*). A graphical overview of the four Intra-modes

is given in Figure 2.6. H.264 supports vertical (Mode 0), horizontal (Mode 1) and diagonal (Mode 3) Intra prediction. In cases with only partially available neighbouring MBs (e.g. border regions), the H.264 standard provides the possibility for using *Mode 2 (DC)* with the available MBs. When there are no available adjacent MBs at all, H.264 specifies the use of *Mode 2 (DC)* using the decimal value *128*.

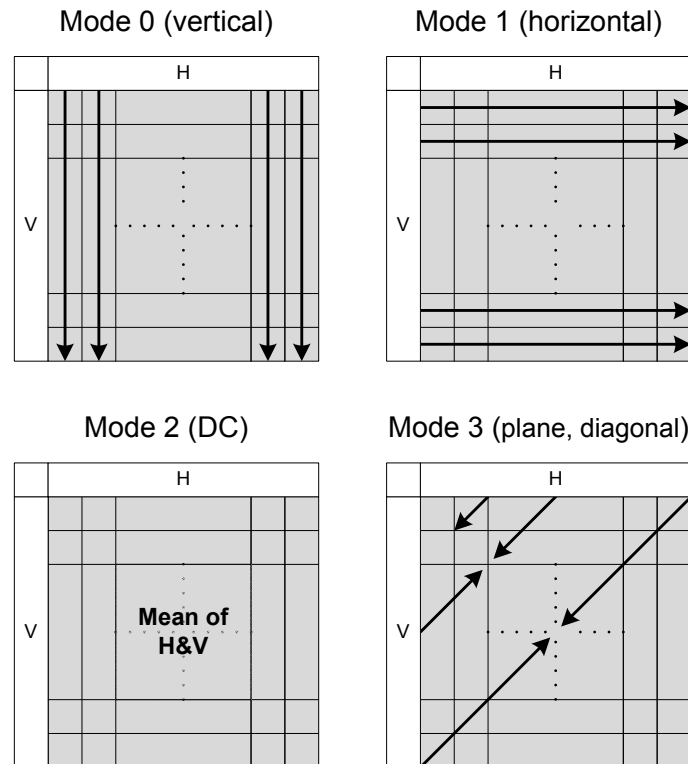


Figure 2.6: Intra 16×16 luminance prediction modes.

Intra- 4×4 luma prediction

Using the 16×16 MBs with the four given coding modes is suitable for homogeneous image-parts with little texture. This is because fewer syntax elements have to be inserted into the data-stream.

However, the partitioning of a MB into sixteen sub-blocks according to Figure 2.7 is preferable when the frame content is finely textured. This partitioning requires a large number of used syntax-elements in the data-stream but also improves the prediction and lowers the prediction error (residuals)¹. The compression of these lowered resid-

¹The prediction error is evaluated using the SAD between the prediction and the original image content.

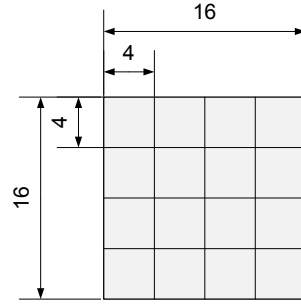


Figure 2.7: Partitioning schema of a 16×16 luma *MB*. Each *MB* consists of 16 4×4 sub-macroblocks (sub-blocks).

uals is in succession more efficient and compensates the additional expenses caused by the higher number of syntax-elements.

The prediction of sub-blocks is similar to the prediction of full *MBs*. The major difference is that five additional coding modes are defined. These coding modes utilise the luminance samples from neighbouring sub-blocks instead of complete *MBs*.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 2.8: Labelling of prediction samples for a 4×4 sub-block.

In Figure 2.8, these samples are labelled as *A-M*. The prediction *P* is determined by the samples labelled *a-p* whereas the calculation investigates the surrounding pixel-values *A-M*.

The Intra coding modes presented in Figure 2.9 can only be used if the necessary border-samples are available. This means that some of the defined coding modes are not applicable at some positions within the frame. A demonstrative example for applying the Intra coding modes is given in Figure 2.10.

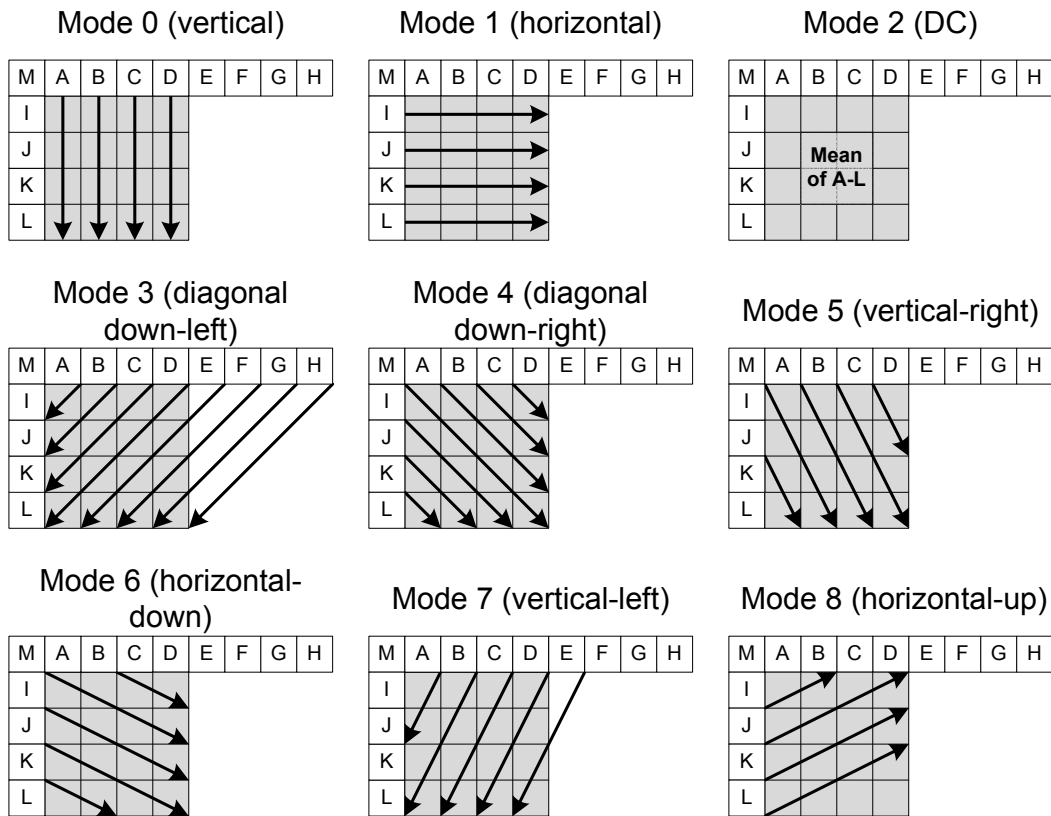


Figure 2.9: Intra 4×4 luminance prediction modes.

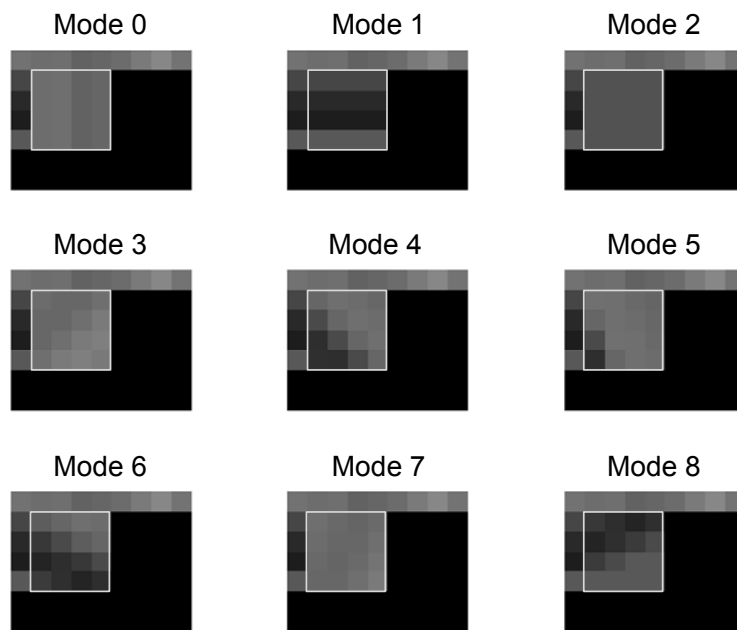


Figure 2.10: Intra 4×4 luminance prediction mode examples.

Intra- 8×8 chroma prediction

The *Baseline Profile* supports progressive scan video formats using the 4:2:0 chroma sub-sampling scheme.

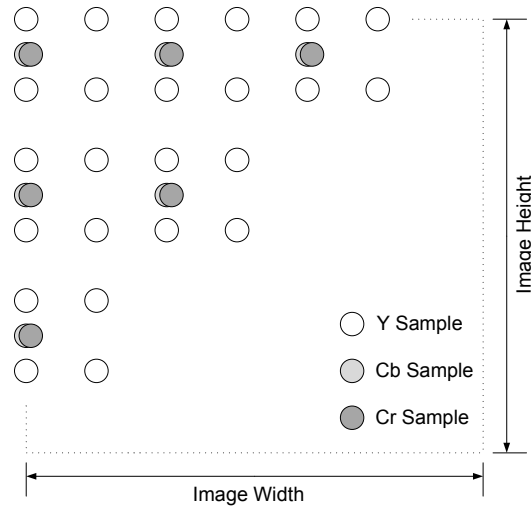
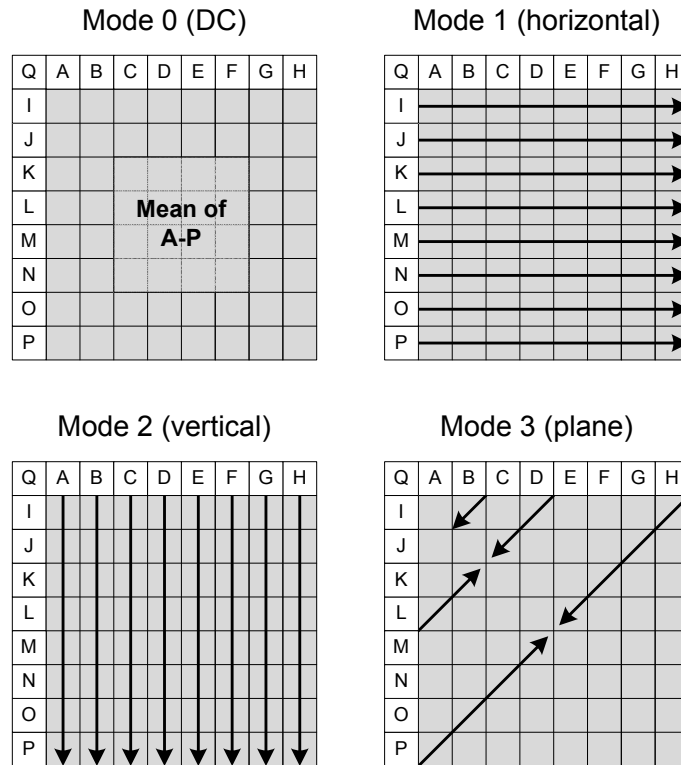


Figure 2.11: 4:2:0 chroma sub-sampling.

Figure 2.11 shows this often used video format which samples the two colour components Cb and Cr in horizontal and vertical direction at half the luminance resolution. This results in a fifty percent data reduction per frame. The 4:2:0 is commonly used for medium quality applications like video conferencing, digital television and storage on mass-media (e.g. digital versatile disc (DVD)). Alternative chrominance formats with higher sampling rates are typically used for studio and high quality applications. For the chrominance components Cb and Cr which are sampled at half the resolution of the luminance channel, the single block size of 8×8 pixels is used for prediction. Like the luma 16×16 prediction, the prediction also uses the vertical (upper) and horizontal (left) neighbour-blocks and supports four different coding modes (see Figure 2.12). H.264 supports a mean (Mode 0), vertical (Mode 1), horizontal (Mode 2) and diagonal (Mode 3) Intra prediction. Depending on the currently available adjacent encoded MBs, only a sub-set of the specified modes are possible. For the first block where no information from neighbouring MBs is available, *Mode-0 (DC)* is used with a constant value of 128 .

Figure 2.12: Intra- 8×8 chrominance prediction modes.

ICMS in H.264

In the H.264 standard, there exist four coding modes for 16×16 MBs and nine coding modes for each sub-block. These coding modes differ in that they use different directions along with the texture pattern which is extrapolated from already encoded pixels. For the chroma MBs, H.264 specifies four coding modes which are similar to the four 16×16 luminance coding modes. An overview of all possible combinations is given in Table 2.1.

Block-size	Number of Modes	Modes to test per MB
16×16 (luma)	4	4
4×4 (luma)	9	144
8×8 (chroma)	4	4

Table 2.1: Overview of the supported Intra coding modes.

Depending on the image content (i.e. texture) of the MB and the used quantisation parameter (QP), a different mode may yield the best prediction. The process of finding the best prediction mode for a MB is called Intra coding mode selection (ICMS).

Typically, this process is computationally very expensive.

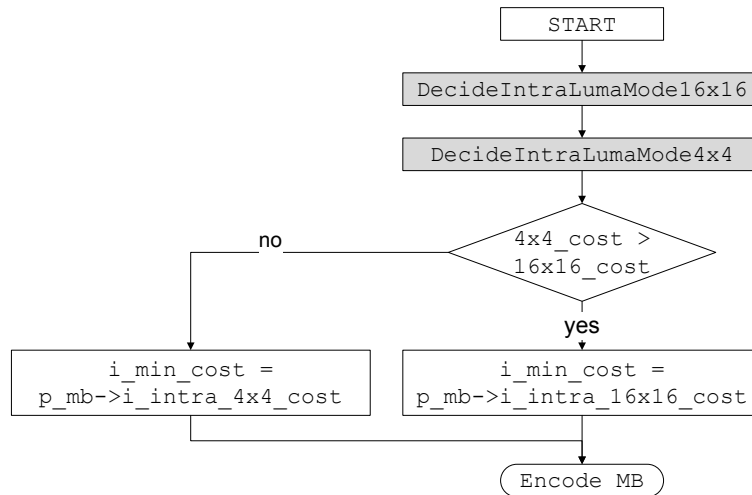


Figure 2.13: Principle work-flow of the used H.264 Intra-mode selection.

Considering the descriptions of Section 2.3, judging the quality of a coding mode (prediction) is done by computing the SAD in intensity between the predicted block and the original uncompressed block. The resulting value is denoted as prediction error or SAD-costs. The Intra coding mode that is finally selected is the one that has the minimum SAD-costs (Figure 2.13).

It is therefore required to calculate the prediction error for each Intra coding mode. This procedure is commonly known as Intra coding mode selection (CMS) and comprises the following steps:

- Determine the best Intra- 16×16 coding mode (4 possibilities).
- Determine the best Intra- 4×4 coding mode (144 possibilities).
- Compare the results of Intra- 16×16 and Intra- 4×4 CMS.

Figure 2.14 shows the average computational complexity of the Intra CMS in comparison to the total encoder's complexity. It is shown that 47% of the required encoding time is used for the Intra CMS. The remaining computation time (53%) is spent on preparing the image-data, transformation of the residuals, entropy coding and writing the compressed information to the NAL. Considering the consumed time of the Intra CMS, it is also shown that nearly 85% of this time is used for deciding the best Intra- 4×4 coding mode, whereas the Intra- 16×16 mode decision consumes just 15% percent.

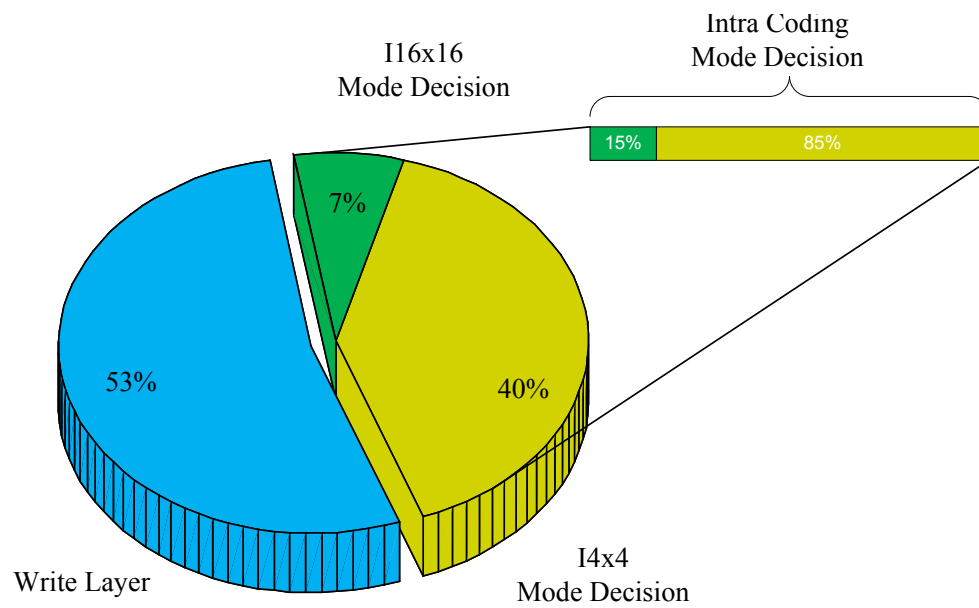


Figure 2.14: Runtime profiling of the encoding process for the 12 test sequences. It is shown that 47% of the computational complexity is caused by the Intra CMS (Intra- 4×4 : 40%; Intra- 16×16 : 7%) and 53% falls upon the remainder.

3 Intra Prediction Analysis

3.1 Test Environment

In this work, a baseline H.264 encoder as well as decoder was provided by ON DEMAND Microelectronics AG (ODM) [25]. It is a commercial encoder optimised for embedded devices. The encoder functionality is used for processing the raw image data, get key figures from the compressed streams and applying new approaches in video processing. The encoder had been extended to provide the required information for a subsequent analysis. This evaluation is carried out by using MathWorks MATLAB [26]. Therefore several scripts and functions have been developed for analysing the data and visualising the results for better understanding and final documentation.

The user specific adaption and debugging of the provided H.264 encoder uses *Microsoft® Visual Studio* as basic development platform. Additionally, some tasks (e.g. code profiling) had to be carried out under Linux using GCC as evaluation environment.

3.2 Quality Metric

For measuring the quality of the compressed sequences, in this thesis the PSNR metric is used. This commonly used metric is derived from the Mean Squared Error (MSE) and rates the pixel prediction of a given frame according to Equation 3.1.

$$MSE = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M [A_{i,j} - B_{i,j}]^2 \quad (3.1)$$

The MSE is specified as the mean squared difference of the undistorted intensity value of a pixel and the degraded intensity value after the lossy data compression (A represents the reconstructed frame and B is the non compressed, original frame.).

$$PSNR = 10 \cdot \log_{10} \left(\frac{(2^n - 1)^2}{MSE} \right) \quad (3.2)$$

Equation 3.2 calculates the PSNR of a single frame. However, it is not allowed to average multiple PSNR-values for determining the mean PSNR of several frames. Instead, it is required to build the non-logarithmic average of the MSE which can then

be inserted into Equation 3.2 (see also [27]).

The PSNR can be computed for the intensity as well as the colour channels. In practice only the luminance is analysed as the human eye is more sensitive to illumination changes compared to chrominance changes.

3.3 Test Sequences

The presented thesis uses a set of 12 video sequences containing different amounts of motion (e.g. camera pan, zoom and translation of objects) and texture.

Sequence	Barcelona	Bus	Canoe	Container
PSNR / dB	33 / 29 / 26	34 / 30 / 27	35 / 31 / 27	36 / 32 / 29
Bits/Pixel	2.1 / 1.9 / 1.7	1.2 / 1.0 / 0.8	1.1 / 0.9 / 0.7	0.8 / 0.6 / 0.5
Sequence	F1Car	Flowergarden	Foreman	Mobile
PSNR / dB	36 / 31 / 28	34 / 30 / 26	36 / 32 / 28	33 / 29 / 25
Bits/Pixel	0.8 / 0.6 / 0.5	1.7 / 1.5 / 1.3	0.6 / 0.5 / 0.4	2.0 / 1.7 / 1.5
Sequence	Mother	Paris	Tempete	Waterfall
PSNR / dB	38 / 34 / 31	35 / 31 / 28	34 / 30 / 26	34 / 30 / 26
Bits/Pixel	0.4 / 0.3 / 0.3	1.2 / 1.0 / 0.8	1.4 / 1.2 / 1.0	1.1 / 0.9 / 0.7

Figure 3.1: Overview of the used test sequences. This figure shows one sample image for each sequence and the individual PSNRs and Bitrates for varying QPs (QP = 30/35/40).

These sequences had been captured in progressive format and CIF resolution at a capturing rate of 25 frames per second. Figure 3.1 shows these sequences along with the key figures PSNR and Bitrate for varying QPs (QP = 30/35/40).

3.4 Motion Analysis of Sequences

The motion analysis presented in this section gives qualitative and quantitative information of the 12 test sequences. This knowledge is useful for understanding the H.264 encoding behaviour.

3.4.1 Metric for the Motion Analysis

For determining the contained sequence motion, it is necessary to calculate the displacement-vectors of corresponding MBs within consecutive frames.

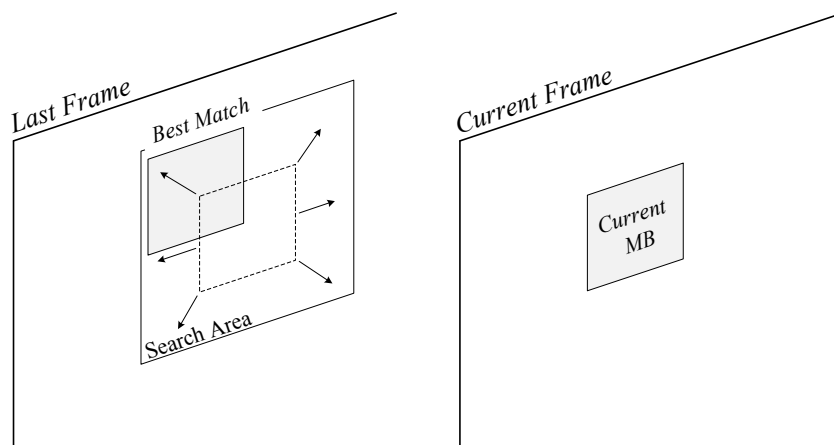


Figure 3.2: Determining the MV by minimising the SAD of corresponding MBs within a given search-area (40×40 pixels).

The calculation scheme is shown in Figure 3.2. The used method thereby evaluates the position of the corresponding MBs inside a search area (40×40 pixels) by minimising the SAD (Equation 3.3 with A and B being the intensity values of the compared image sections at the two consecutive timestamps t and $t-1$) of candidate MBs at various locations.

$$SAD = \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} |A_{(i,j)} - B_{(i,j)}| \quad (3.3)$$

To characterise the content motion for each test-sequence, the motion-vector (MV)s of the contained MBs have been determined, classified and graphically presented. According to the length of the MB's MV, the MBs are arranged into six different classes.

Class No.	Description	Definition
1	Pure Static	$ MV_{lim} = 0 \frac{pixel}{frame}$
2	Very Slow Motion	$ MV_{lim} \leq 1.4 \frac{pixel}{frame}$
3	Slow Motion	$ MV_{lim} \leq 4.2 \frac{pixel}{frame}$
4	Moderate Motion	$ MV_{lim} \leq 5.7 \frac{pixel}{frame}$
5	High Motion	$ MV_{lim} \leq 11.3 \frac{pixel}{frame}$
6	Very High Motion	$ MV_{lim} \leq 17.0 \frac{pixel}{frame}$

Table 3.1: Definition of the classification-groups used for the MV-distribution.

The grouping of the six classes presented in Table 3.1 takes place by ranking the MVs caused by equal horizontal and vertical movements.

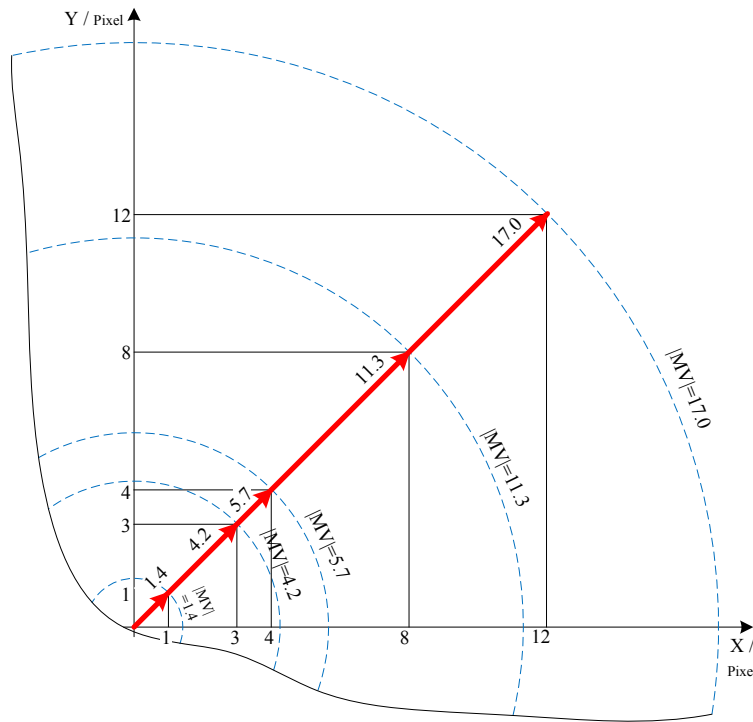


Figure 3.3: Determining the norm of the limiting MVs by using the vertical (y) and horizontal (x) pixel-displacements.

For calculating the given values ($0, 1.4, 4.2, 5.7, 11.3$ and 17.0), *Pythagoras' theorem* is used with the horizontal and vertical movements (see Figure 3.3).

3.4.2 Motion Analysis

In this section, 12 sequences are presented and analysed with respect to their contained motion. These sequences are used in the following sections.

Barcelona

This colourful sequence is strongly textured throughout each frame (see Figure 3.1). The dominant motion is caused by a slow camera zoom-out. However, as shown in Table 3.2, nearly sixty percent of the scene is purely static ($|MV| = 0 \frac{\text{pixel}}{\text{frame}}$). The remaining forty percent of the content has a spatial motion of $\leq 1.4 \frac{\text{pixel}}{\text{frame}}$ (one pixel in vertical and horizontal direction).

Bus

This sequence has a medium frame texture and uniform motion in the foreground and background. The motion is caused by a moving omnibus (driving to the left border) and a camera pan for tracking the bus. Considering the motion information shown in Table 3.2, the average motion is below $11.3 \frac{\text{pixel}}{\text{frame}}$. The motion information also shows two transitions. The first one at frame 28 and the second at frame 79. These transitions indicate motion changes caused by the complex overlay of the camera pan and the moving bus, whereas the camera pan causes higher motion.

Canoe

This high motion sequence shows a canoe in a fast streaming river (see Figure 3.1). The contained motion (Table 3.2) is caused by interfering a horizontal camera pan and the flow of the water. In the middle of the sequence, the direction of the camera pan changes. The high motions are causing a motion-blur, resulting in a weak sequence texture.

Container

This low textured video sequence has an almost static content ($|MV| = 0 \frac{\text{pixel}}{\text{frame}}$). It presents a slowly moving container-ship on the shore (see Figure 3.1). Table 3.2 shows that the motion of the ship is just about $1.4 \frac{\text{pixel}}{\text{frame}}$.

F1 car

This low textured sequence has various camera viewpoints with additional changing camera pan for tracking a fast moving formula one car (Figure 3.1). The contained motion measures approximately $5.7 \frac{\text{pixel}}{\text{frame}}$ on average and is shown in Table 3.2. The

texture of the foreground road surface in the first 50 frames is fairly low compared to the remaining parts of the scene (e.g. car). By moving the camera down, the amount and value of the road texture increases. This camera pan also causes a higher sequence motion which is noticeable from frame 40 to frame 55. In this range, almost two thirds of the sequence motion is between $5.7 \frac{\text{pixel}}{\text{frame}}$ and $11.3 \frac{\text{pixel}}{\text{frame}}$.

Flowergarden

This video sequence contains a uniform, slow horizontal motion caused by a moving camera. The sky in the background has no distinct texture while the flower covered foreground has a clear texture-pattern with high intensity contrast (see Figure 3.1). These intensity-differences are causing higher prediction errors (residuals), leading to increased data-rates.

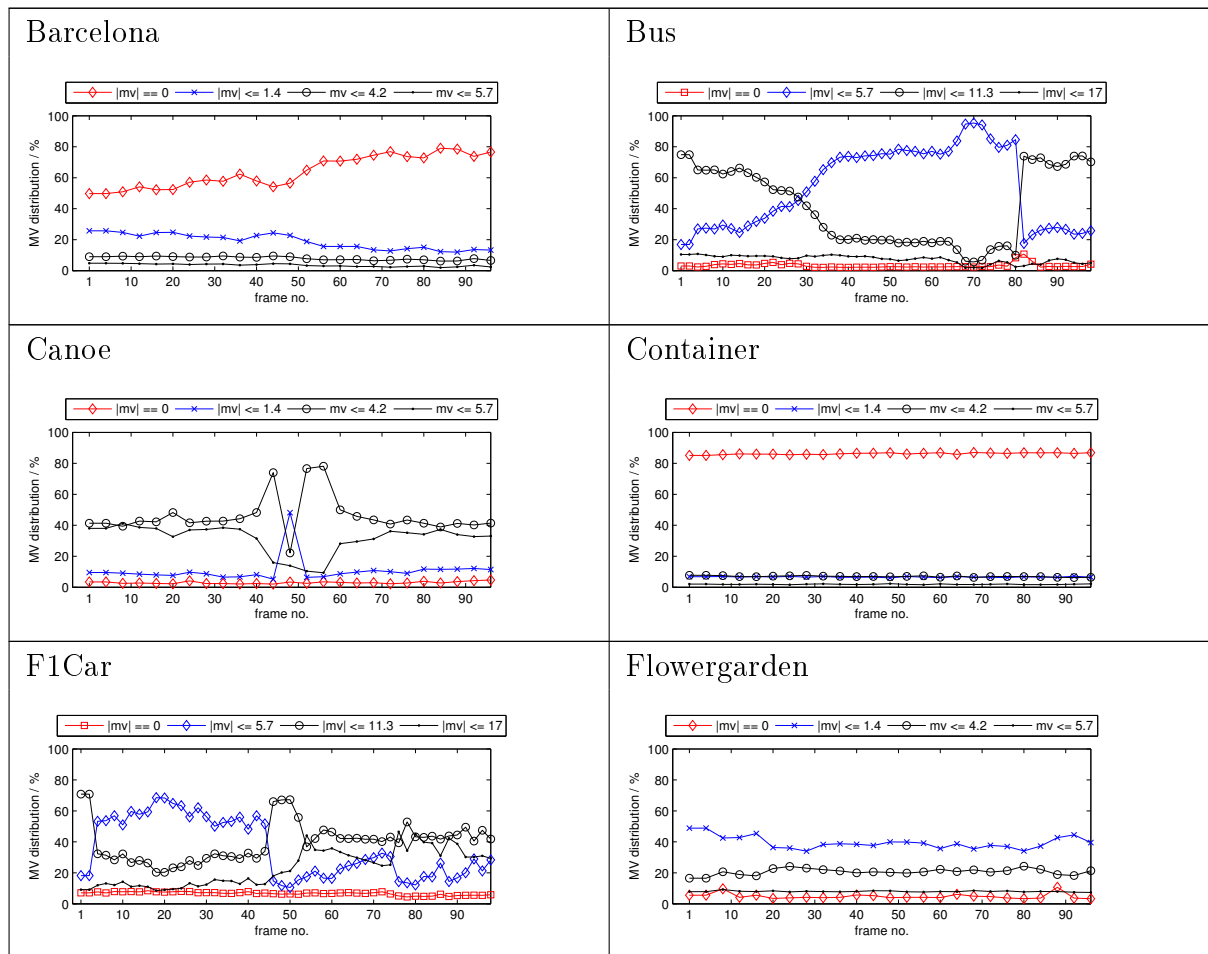


Table 3.2: Classified MVs for the test sequences barcelona, bus, canoe, container, f1car and flowergarden. Each resulting curve shows the measured values for the first 100 frames.

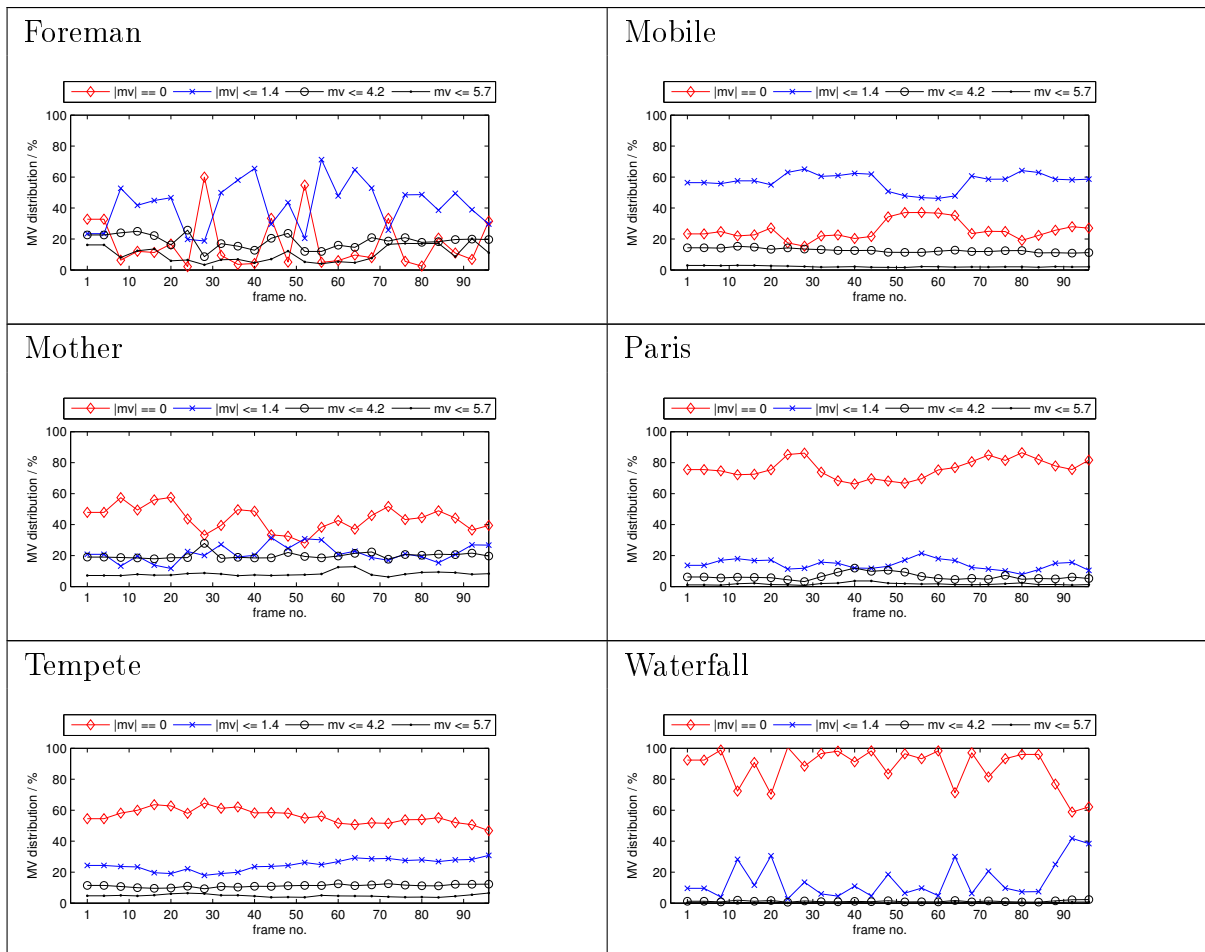


Table 3.3: Classified MVs for the test sequences foreman, mobile, mother, paris, tempete and waterfall. Each resulting curve shows the measured values for the first 100 frames.

Foreman

This low textured sequence shows the head of a building site foreman, captured by a non-professional hand-operated camera in the course of an interview (Figure 3.1). The contained motion is mainly caused by the foreman's shaking head (in the first part of the sequence) and a camera pan (in the second part of the sequence). In addition, irregular jitter-movements caused by cameraman's hand-operation are superimposed. As shown in Table 3.3, the average motion is about $1.4 \frac{\text{pixel}}{\text{frame}}$.

Mobile & Calendar

This sequence contains large moving image partitions. The motion is mainly caused by a horizontally driving model railway and a vertically moving calendar (Figure 3.1).

The average motion measures approximately $1.4 \frac{\text{pixel}}{\text{frame}}$ (Table 3.3).

Mother

In this video clip a mother is shown with her child. Both are sitting still in the foreground (Figure 3.1). The achieved data-rates are very low as the sequence contains insignificant texture patterns. Considering the motion information from Table 3.3, it is obvious that the average MV is below $1.4 \frac{\text{pixel}}{\text{frame}}$.

Paris

This sequence shows a video conference, captured with a statically mounted camera. The recorded scene shows a room with non-moving furniture and two moving persons (Figure 3.1). The colourful frames have a sharp contrast with large homogeneous areas. The contained motion is caused by the moving persons. However, as shown in Table 3.3, the scene is mostly static.

Tempete

The captured scene in this sequence shows a mountain in the background and a colourful flower in the foreground (Figure 3.1). The leaves of the brightly coloured flower are swaying irregularly as there is a stormy wind. These jitter-movements are overlaid by a slow zoom-out and a slight camera pan. The resulting MV is below $1.4 \frac{\text{pixel}}{\text{frame}}$ on average (Table 3.3).

Waterfall

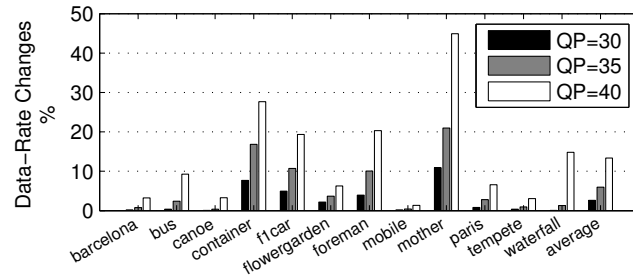
This sequence shows a highly textured forest and a waterfall in the centre (Figure 3.1). The contained motion is caused by a slow camera zoom-out which is focused on the waterfall. Nearly 90 percent of the contained MVs are below $1.4 \frac{\text{pixel}}{\text{frame}}$ (Table 3.3).

In the following sections, the impacts on the quality and the data-rate are analysed when using either 4×4 or 16×16 Intra-coding modes.

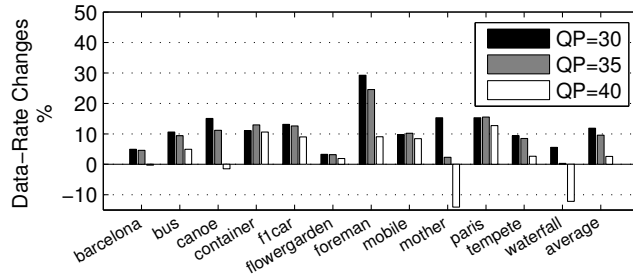
3.5 Analysing the Impact of the Block Size

The aim of this section is to evaluate the coding efficiency of the Intra- 16×16 and Intra- 4×4 prediction modes. For determining the coding performance of these coding techniques, the achieved PSNRs and data rates are determined and afterwards compared for the following three encoding conditions:

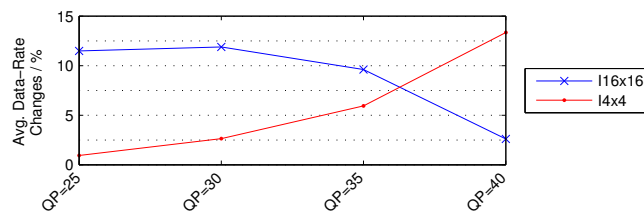
- Full Intra-mode encoding.
- Intra- 16×16 only encoding.
- Intra- 4×4 only encoding.



(a)



(b)



(c)

Figure 3.4: Data-rate changes for Intra-only encoding using either 4×4 or 16×16 coding modes. The data-rates are compared to the values of a full Intra-mode encoding. (a) Intra- 4×4 only encoding. (b) Intra- 16×16 only encoding. (c) Average data-rate changes for four different QPs (25, 30, 35 and 40) and Intra coding modes.

The resulting data-rates of the full Intra-mode encoding are used as reference values for comparing the data-rates of the Intra- 16×16 only encoding and Intra- 4×4 only encoding, respectively. This comparison is given in Figure 3.4, presenting the average data-rate changes for the 12 test-sequences using different coding modes and different QPs.

Figure 3.4(a) shows that for lower QPs (i.e. better image quality) the Intra- 4×4 encoding is more efficient on average. Using the Intra- 16×16 coding modes is more beneficial for higher QPs (lower image quality, see Figure 3.4(b)). Using lower QPs to maintain fine image structures also requires smaller block sizes to produce accurate predictions (smaller residuals). So, the achievable data-rate is lower when using Intra- 4×4 in this case. In contrast, a decrease of the image quality eliminates fine structures, making 16×16 coding modes more attractive as they are able to produce comparable residuals in this case while requiring fewer syntax elements in the compressed data-stream for inserting the coding mode information.

Figure 3.4(c) illustrates the summary of the above given interrelation and shows the favoured operational area of Intra- 4×4 and Intra- 16×16 coding modes with respect to the achievable data-rates. The plotted curves are showing the average data-rate changes for Intra- 4×4 or Intra- 16×16 coding modes and varying QPs. It is shown that for QPs below 37, the Intra- 4×4 coding modes achieve lower data-rates compared to the Intra- 16×16 coding modes as the smaller block-sizes are better suited for maintaining fine image structures and keeping the residuals low. These fine image structures disappear for higher QPs and the Intra- 16×16 coding modes are more advantageous as less syntax elements are required.

Figure 3.5 presents the impacts of the coding block-size on the achieved PSNR values for varying QPs (30, 35 and 40). The plotted graph shows the absolute PSNR differences between the sequences using full Intra encoding, Intra- 4×4 only encoding (Figure 3.5(a)) and Intra- 16×16 only encoding (Figure 3.5(b)).

It is shown that the average PSNR differences are in both cases in the range between 0 - 0.8dB. However, the PSNR differences for of the Intra- 4×4 coding modes (Figure 3.5(a)) are lower compared to the Intra- 16×16 coding modes (Figure 3.5(b)). This difference increases for higher QPs, which is illustrated in the summarising graph in Figure 3.6. This behaviour is mainly due to the better encoding performance of smaller block-sizes and the higher number of coding modes.

3.5 Analysing the Impact of the Block Size

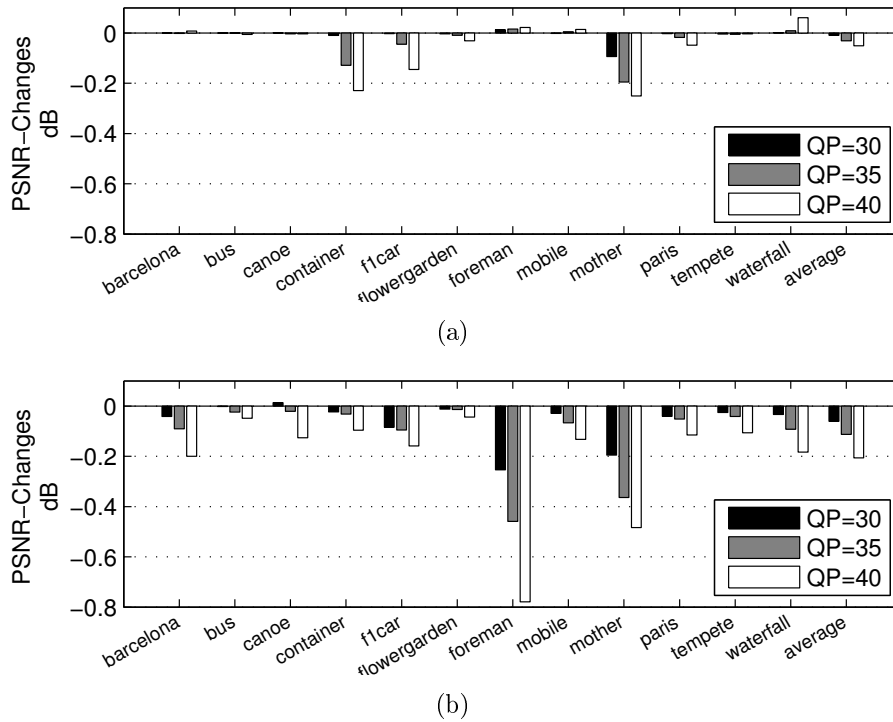


Figure 3.5: PSNR-Changes for Intra-only encoding using either 4×4 or 16×16 coding modes. (a) PSNR-Changes for Intra- 4×4 only encoding. (b) PSNR-Changes for Intra- 16×16 only encoding.

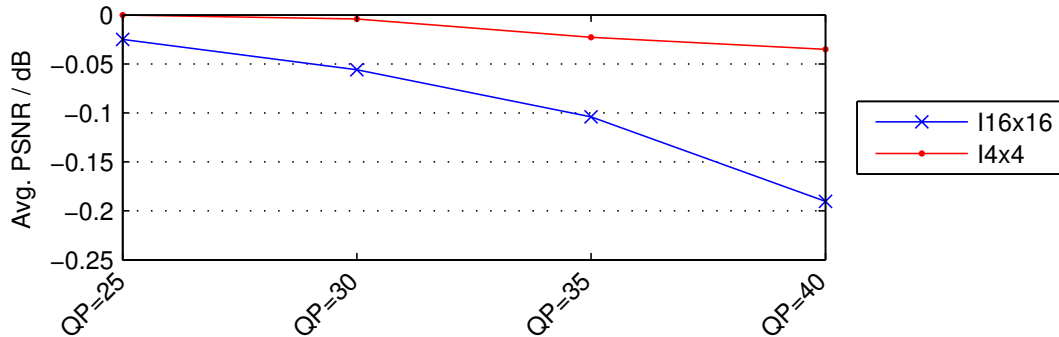
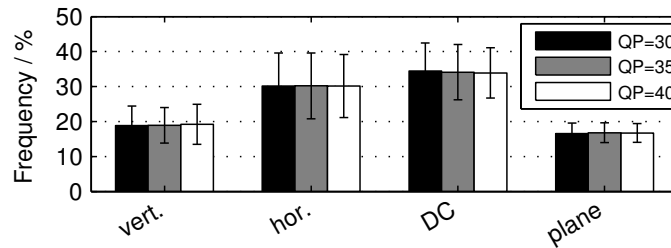


Figure 3.6: Summary of the PSNR changes for Intra-only encoding using either 4×4 or 16×16 coding modes for various QPs (25, 30, 35 and 40) and coding modes. The data-rates are compared to the values of a full Intra-mode encoding.

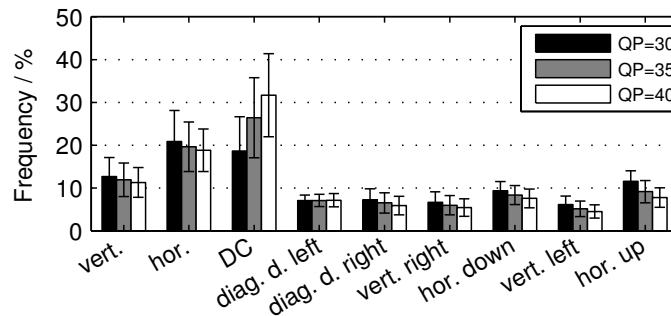
3.6 Reduced Number of Intra-Coding Modes

A high number of coding modes results in better coding efficiency but increases the computational complexity as more SAD-calculations are required. The aim of this section is to lower the required complexity by omitting computational expensive modes that do not improve the coding efficiency significantly.

The following analyses therefore evaluate the frequency and the impact on the coding efficiency of the coding modes of sub-blocks and MBs. This allows the identification of essential and insignificant coding modes and how they affect the coding efficiency.



(a)



(b)

Figure 3.7: Average frequency of the coding modes for various QPs and the 12 test-sequences. (a) Frequency of the Intra- 16×16 coding modes. (b) Frequency of the Intra- 4×4 coding modes.

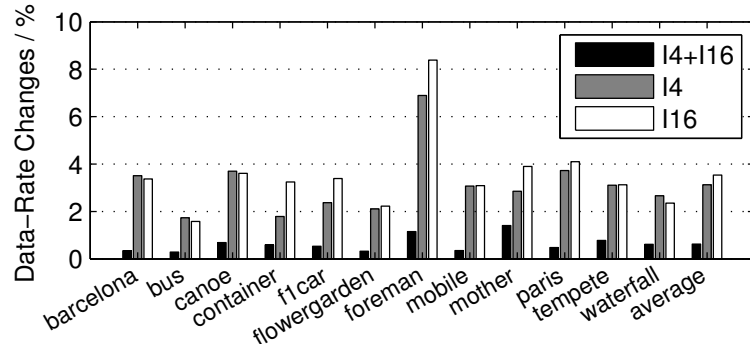
The evaluation results of the coding mode frequencies are given in Figure 3.7. The two histograms in Figure 3.7 are showing the coding mode frequencies for the four Intra- 16×16 coding modes (Figure 3.7(a)) and the nine Intra- 4×4 coding modes (Figure 3.7(b)) over all sequences.

Figure 3.7(a) shows that depending on the used QP, the modes *DC*, *H*, *V* occupy nearly 85 percent on average when using Intra- 16×16 coding modes.

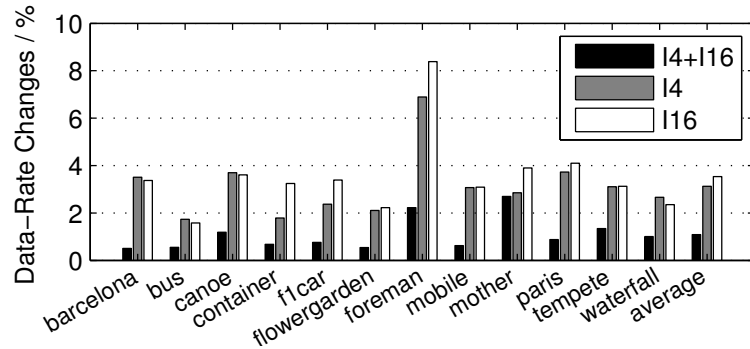
For the 12 test sequences, using the simple modes only affects 15 percent of the

available MBs and considerably lowers the computational complexity as fewer SAD-calculations are required for the CMS.

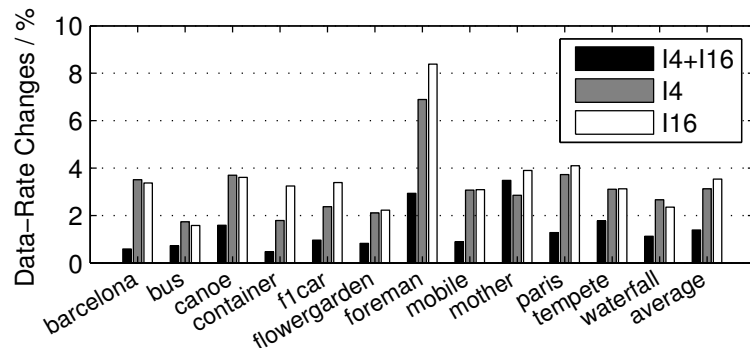
A similar result for Intra- 4×4 sub-blocks is presented in Figure 3.7(b). It is shown that nearly 60 percent of the encoded sub-blocks are using the DC , H or V mode. The remaining 40 percent are equally distributed over the other coding modes.



(a)



(b)



(c)

Figure 3.8: Relative Data-Rate changes for a reduced number of Intra coding modes. The figures show the data-rate changes for the 12 test sequences for three QPs. (a) Result for QP=30. (b) Result for QP=35. (c) Result for QP=40.

The second part of this section analyses the negative effects when omitting the complex Intra coding modes. Therefore, the provided H.264 encoder has been modified for only using the Intra coding modes V , H and DC . The following evaluations analyse the 12 test sequences using three test-cases. In the first test-case, the encoder uses 4×4 and 16×16 blocks with a reduced number of Intra coding modes (i.e. H, V and DC only). In the second and third test-case, the 4×4 and 16×16 blocks are used respectively.

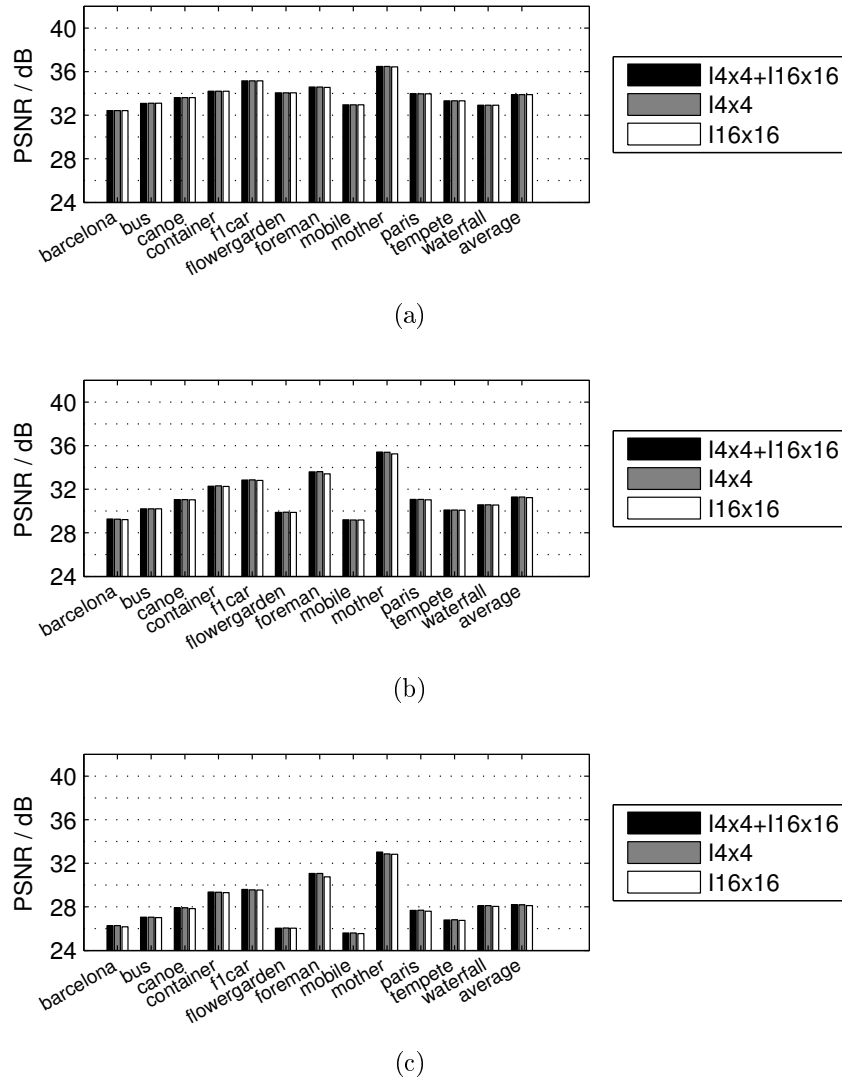


Figure 3.9: PSNR changes for a reduced number of Intra coding modes and the 12 test sequences. (a) Result for QP=30. (b) Result for QP=35. (c) Result for QP=40.

The result of this evaluation is shown in the Figures 3.8 and 3.9. The given values for the data-rates in Figure 3.8 are compared to the results of a full Intra coding mode

selection and using three QPs (30, 35 and 40).

Figure 3.8(a) shows that the data-rate changes for QP=30 are below three percent on average for the three test-cases. These values are almost not affected by the used QP, as the average data-rate changes for QP=35 and QP=40 are still below three percent (see Figures 3.8(b) and 3.8(c)).

Considering the PSNR-values presented in Figure 3.9, no significant changes are measurable when reducing the number of coding modes. The influence of the used QP is also very low.

The presented results show that a reduced number of Intra coding modes causes slightly increased bit-rates while maintaining the image quality. This means that H.264 is able to compensate prediction errors by increasing the data-rate. The benefit, however, is a considerable decrease of the computational complexity.

3.7 Intra-Only Encoding of Static Regions

This section analyses the encoding behaviour of the Intra- 4×4 coding modes for static regions in a sequence. In the context of this work, a region is static if no motion occurs at this frame-position over the time.

The aims for the planned analysis are to evaluate the validity of the three assumptions (A1-A3) stated below:

- A1** Static sub-blocks tend to retain their coding mode between consecutive frames (i.e. the most efficient coding mode typically outperforms the other modes significantly).
- A2** A sub-block that retains its coding mode over time will also have similar SAD-costs.
- A3** Spatially adjacent sub-blocks (e.g. the left and upper sub-block) tend to have a similar temporal coding behaviour.

These assumptions are derived from the following theoretic considerations. Except of sensor-related effects, the changes in a sub-block are typically too small for causing a change of the coding modes (A1). As a static sub-block should not change significantly over time, the expected SAD-costs should also remain nearly constant (A2).

For the Intra- 4×4 coding mode selection it is useful to exploit the fact that spatially adjacent pixels in a picture are typically correlated. This implies that neighbouring sub-blocks often contain similar texture patterns and can be efficiently described by similar Intra coding modes. Assumably, there is also a high probability that static sub-blocks will also have static neighbours (A3).

3.7.1 Mode Retention of Static Regions

The major goal of this section is to evaluate the temporal coding mode changes of static sub-blocks as the gained conclusions and interrelations are important for understanding the Intra- 4×4 encoding of slow motion scenes. In this thesis, a sub-block is static when its maximum translation is one pixel in horizontal and/or vertical direction between two consecutive frames.

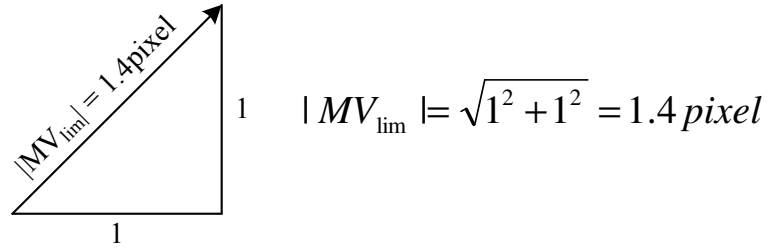


Figure 3.10: Determining the magnitude of the limiting Motion Vector using the vertical and horizontal components.

For computing the magnitude of the limiting Motion Vector MV_{lim} (1.4 pixel) Pythagoras' theorem is used (Figure 3.10).

Detecting static sub-blocks for this analysis therefore requires adequate motion information. As the Intra encoding does not provide any motion information, it is necessary to determine the MVs separately. This information is used for analysing the mode retention of static regions at the three conditions C1-C3:

$$\mathbf{C1} \quad (\text{old mode} == \text{new mode}) \ \&\& \ |MV| \leq MV_{lim}$$

$$\mathbf{C2} \quad (\text{old mode} != \text{new mode}) \ \&\& \ |MV| \leq MV_{lim}$$

$$\mathbf{C3} \quad (\text{old mode} == \text{new mode}) \ \&\& \ |MV| > MV_{lim}$$

Condition *C1* describes static sub-blocks with constant Intra-mode and *C2* describes static sub-blocks with changing Intra-modes. Condition *C3* covers non-static sub-blocks with unchanged Intra-modes.

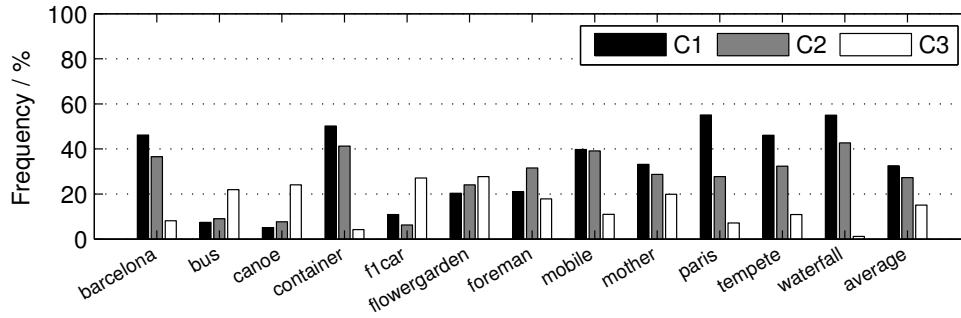


Figure 3.11: Interrelation of the mode retention and image motion for the 12 test sequences and QP = 30.

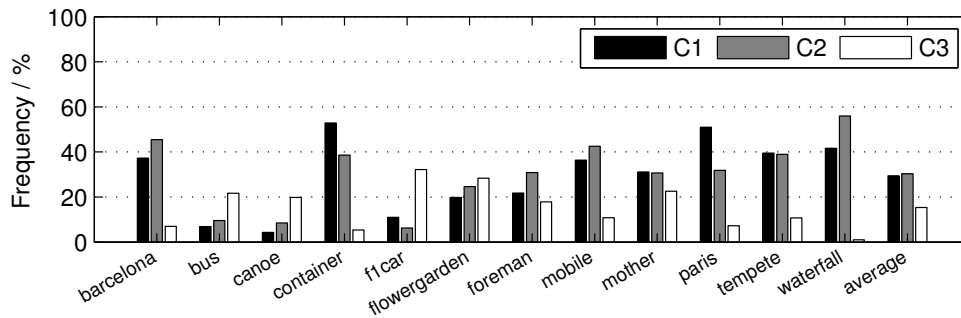


Figure 3.12: Interrelation of the mode retention and image motion for the 12 test sequences and QP = 35.

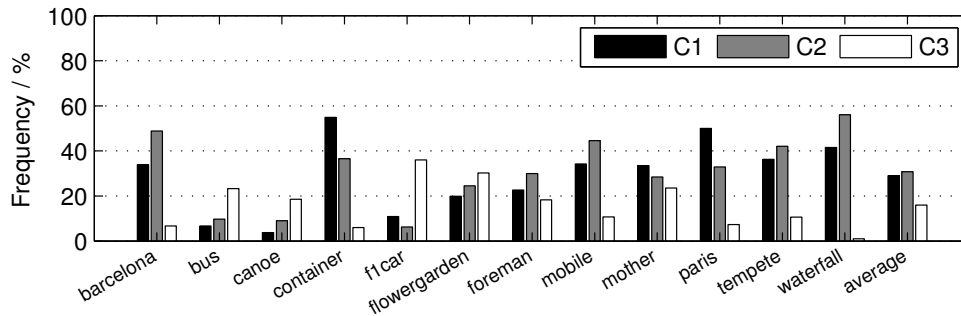


Figure 3.13: Interrelation of the mode retention and image motion for the 12 test sequences and QP = 40.

The Figures 3.11, 3.12 and 3.13 show the evaluation results of these conditions for various QPs. Each bar represents the frequency of one condition in relation to the total amount of sub-blocks.

It is shown that sequences with higher motion tend to have low values for condition *C1* and *C3* as there are only few static areas. Furthermore, images with uniform texture tend to have higher values of *C3*.

When comparing the quantities of the condition $C1$ and $C3$ it is obvious that approximately 50 percent of the static sub-blocks tend to change their coding mode. This is interesting, as intuitively it should not be the case that static sub-blocks change their coding mode. However, the reason is that some modes produce very similar SAD-costs and depending on the adjacent neighbours, the ranking of the most probable coding mode may change. Furthermore, it can be seen that this characteristic is barely affected by the used QP.

Considering the average influence of QP, the given figures show that $C1$ for QP=35 is about five percent higher compared to QP=30 or QP=40. For $C2$ and $C3$ now considerable influence of QP is noticeable.

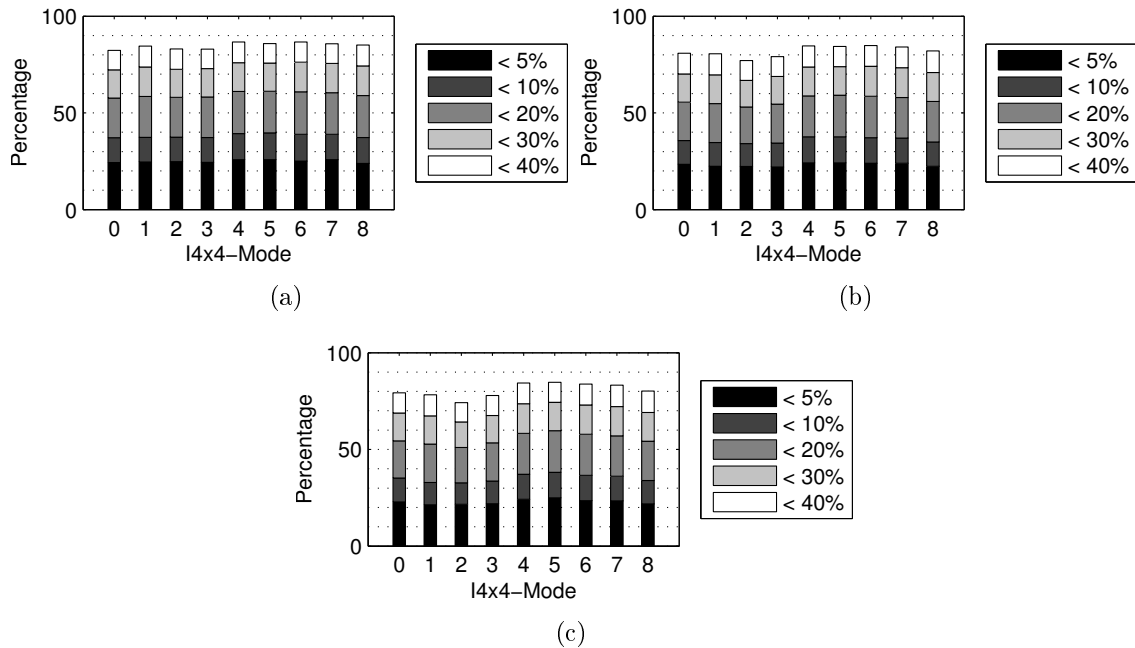


Figure 3.14: SAD-cost changes of static sub-blocks with alternating coding modes. The changes are grouped in five classes, analysed in terms of frequency. (a) SAD-cost changes for QP=30. (b) SAD-cost changes for QP=35. (c) SAD-cost changes for QP=40.

Figure 3.14 illustrates the SAD-cost changes of static sub-blocks with changing coding modes. The examined plots show the temporal changes of the SAD-costs for each static sub-block. These SAD-cost changes are grouped in five classes. The first class shows the frequency of the relative SAD-cost changes equal to or less than 5%, the second class between 5% and 10% and so on. This statistical analysis exists for each of the nine coding modes.

It is shown that nearly 60% of these sub-blocks have SAD-cost changes of less than

20%. This means that irrespective of the used coding mode the SAD-costs remain stable. These observations show that assumption *A1* is valid for a larger part of the static sub-blocks.

3.7.2 Temporal Changes of the SAD-Costs

For investigating assumption *A2* it is necessary to evaluate the temporal SAD-cost changes of co-located sub-blocks within consecutive frames. Figure 3.15 shows the statistical evaluation of the SAD-costs derived from the 12 test-sequences. This statistical evaluation is expressed as the ratio between the SAD-cost of two co-located sub-blocks. Equation 3.4 shows the calculation of the ratio SAD_{ratio} for two co-located sub-blocks at the position (x, y) and the timestamps $t - 1$ and t .

$$SAD_{ratio} = \frac{SAD(x, y)_t}{SAD(x, y)_{t-1}} \quad (3.4)$$

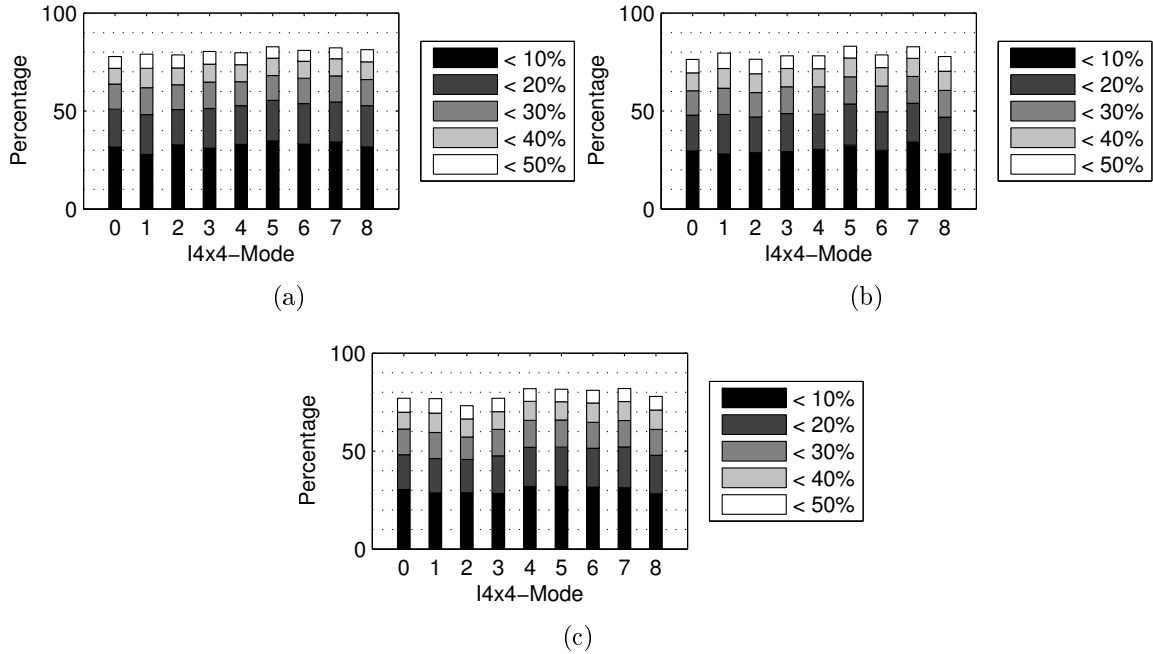


Figure 3.15: SAD-cost changes of static sub-blocks with unchanged coding mode. The changes are grouped in five classes, analysed in terms of frequency. (a) SAD-cost changes for QP=30. (b) SAD-cost changes for QP=35. (c) SAD-cost changes for QP=40.

The stacked bars in Figure 3.15 show the SAD-cost changes of static sub-blocks with unchanged coding mode. The changes are grouped in five classes, analysed in terms

of frequency. The first class shows the frequency of SAD-cost changes equal to or less than 5%, the second class between 5% and 10% and so on.

It is shown that 50 percent of the static sub-blocks tend to have SAD-ratios above 20 percent and 80 percent of the static sub-blocks are within the 50 percent ratio-limit. This result indicates that assumption *A2* is valid for more than half of the static sub-blocks in the test sequences. For 65 percent of the sub-blocks, the changes are below 50 percent.

3.7.3 Correlation Between Neighbouring Static Sub-Blocks

For examining assumption *A3*, the coding mode of each constant sub-block is compared with the coding modes of its left and upper neighbour sub-block. The statistical probability for each mode combination of two spatial neighbours is analysed.

For calculating the correlation between the coding mode of the current sub-block and the left/upper sub-block, four sequences have been used. This ensures an independent verification when using the heuristic described in Section 5 with the 12 test-sequences. These four sequences are *carphone* (high motion, medium texture), *claire* (low motion, low texture), *coastguard* (high motion, high texture) and *news* (medium motion, medium texture) (Figure 3.16).





Sequence	Carphone	Claire	Coastguard	News
				
PSNR / dB	33 / 29 / 26	38 / 34 / 30	33 / 29 / 25	36 / 32 / 29
Bits / Pixel	1.4 / 1.2 / 1.0	0.8 / 0.6 / 0.4	1.6 / 1.4 / 1.2	1.0 / 0.8 / 0.6
Motion	medium	low	high	low
Texture	medium	low	high	medium

Figure 3.16: Four extra sequences at CIF resolution using Intra- 4×4 encoding and QP=30/35/40. Rows 2 and 3 show the PSNR values and the resulting data-rate. In Row 4, the sequence motion is described. The last row gives qualitative statements to the contained texture.

The matrix in Figure 3.17 shows the statistical correlation of the following three sub-blocks: current, left and upper. Each field in the matrix shows the most probable coding mode for a given combination of the left and upper neighbour. The second value in each field presents the evaluated probability for this correlation.

It is shown that the coding modes of neighbouring sub-blocks are only weakly correlated. For most cases the best coding mode for the current sub-block can be predicted with a certainty of 25 to 30 percent.

However, if the left or upper sub-blocks are using the same coding modes (in the range of 0 to 2), the current sub-block uses the same coding mode with a probability of about 65 percent. This implies that assumption $A3$ is only valid under certain conditions. To use this optimisation heuristics it is necessary to also consider these conditions.

		Mode of upper 4x4-block									
		0	1	2	3	4	5	6	7	8	
Mode of left 4x4-block	0	0 (65%)	0 (35%)	0 (35%)	0 (30%)	0 (35%)	0 (40%)	0 (35%)	0 (40%)	0 (35%)	
	1	1 (40%)	1 (65%)	1 (40%)	1 (35%)	1 (35%)	1 (35%)	1 (40%)	1 (40%)	1 (45%)	
	2	2 (30%)	1 (35%)	2 (65%)	2 (30%)	2 (25%)	2 (30%)	1 (30%)	2 (35%)	1 (30%)	
	3	0 (25%)	1 (30%)	2 (25%)	2 (25%)	2 (25%)	2 (30%)	1 (25%)	2 (25%)	8 (25%)	
	4	0 (25%)	1 (30%)	2 (25%)	1 (25%)	6 (25%)	0 (25%)	1 (25%)	0 (25%)	1 (25%)	
	5	0 (30%)	1 (25%)	0 (25%)	0 (25%)	0 (25%)	5 (30%)	1 (25%)	0 (25%)	8 (25%)	
	6	0 (25%)	1 (35%)	1 (25%)	1 (25%)	1 (25%)	1 (25%)	6 (35%)	1 (25%)	1 (30%)	
	7	0 (30%)	1 (25%)	2 (25%)	7 (25%)	0 (25%)	0 (25%)	1 (25%)	7 (25%)	8 (25%)	
	8	1 (25%)	1 (35%)	1 (30%)	1 (25%)	1 (30%)	1 (25%)	1 (30%)	1 (30%)	1 (30%)	

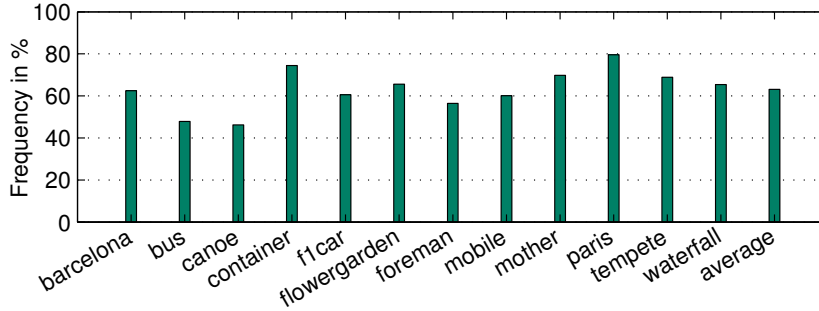
Figure 3.17: Matrix showing the prediction-heuristic. The values in the array are the most likely modes and the probability values.

The second issue of assumption $A3$ refers to the notion that a static sub-block will reuse its coding mode in case that adjacent neighbouring sub-blocks are keeping their coding modes. The following evaluation therefore examines the frequency of the three cases below:

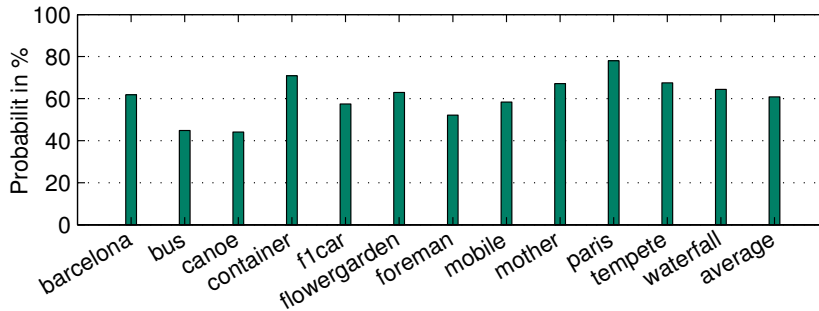
$$\text{Case 1: reuse coding mode} = \begin{cases} \text{true if left sub-block reused mode} \\ \text{false otherwise} \end{cases}$$

$$\text{Case 2: reuse coding mode} = \begin{cases} \text{true if upper sub-block reused mode} \\ \text{false otherwise} \end{cases}$$

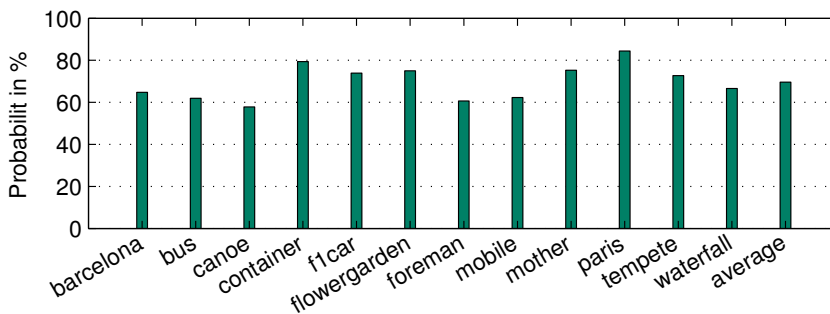
$$\text{Case 3: reuse coding mode} = \begin{cases} \text{true if left and upper sub-block reused mode} \\ \text{false otherwise} \end{cases}$$



(a)



(b)



(c)

Figure 3.18: Probability of the coding mode reuse for sub-blocks where the left and/or upper neighbours keep their coding mode. (a) Evaluation result exploiting the left neighbour (Case 1). (b) Evaluation result exploiting the upper neighbour (Case 2). (c) Evaluation result exploiting the left and upper neighbours (Case 3).

Figure 3.18 shows the evaluation result of the three cases when using the 12 test sequences. The evaluation result covers the probability of the coding mode reuse of sub-blocks where the left and/or the upper neighbours also keep their coding modes.

It is shown that around 60 percent of the evaluated sub-blocks tend to reuse their

coding mode when either the left or the upper neighbour leaves its coding mode unchanged (Figure 3.18(a) and 3.18(b)).

The result for case 3 is shown in Figure 3.18(c). The average probability is approximately 70 percent when the coding modes of both neighbours are unchanged.

Considering this result, assumption *A3* of Section 3.7 can be deemed as correct for a large number of sub-blocks.

3.7.4 Résumé

The goal of this section was to analyse the encoding behaviour of static sub-blocks. It was therefore necessary to evaluate the temporal SAD-cost and coding mode changes according to theoretic considerations (assumptions *A1*, *A2* and *A3*).

Section 3.7.1 showed that a large part of static sub-blocks tend to reuse the coding modes of the previous frame. Coding mode changes for static sub-blocks are mostly caused by similar SAD-cost of multiple coding modes. This results in a similar ranking of the best coding modes and may change the mode of a static sub-block between two frames.

A large number of sub-blocks tend to maintain their coding modes and SAD-costs whereas others are vulnerable of changing both the coding mode and SAD-costs. The evaluation of Section 3.7.3 figured out the spatial dependencies of the coding mode reuse considering the left and upper neighbours. This analysis showed that sub-blocks tend to maintain their coding modes when the left and/or upper neighbours are static.

4 Single Criterion Coding Mode Reuse (SCCMR)

This section demonstrates the development of a criterion for deciding if the previous coding mode of a sub-block is reused without performing a computational expensive evaluation of all coding modes (i. e. full mode search). Based on the conclusions of Section 3, it can be assumed that the SAD-costs of sub-blocks provide a good measure for the ‘stability’ of a sub-block’s coding mode. The decision whether to reuse the previous coding mode is based on the comparison of the SAD-cost of two co-located sub-blocks according to following the equation:

$$SAD_{ratio} = \left| 1 - \frac{SAD_t}{SAD_{t-1}} \right| \quad (4.1)$$

In this equation SAD_t and SAD_{t-1} are the sub-block’s SAD values at frame t and $t - 1$, respectively.

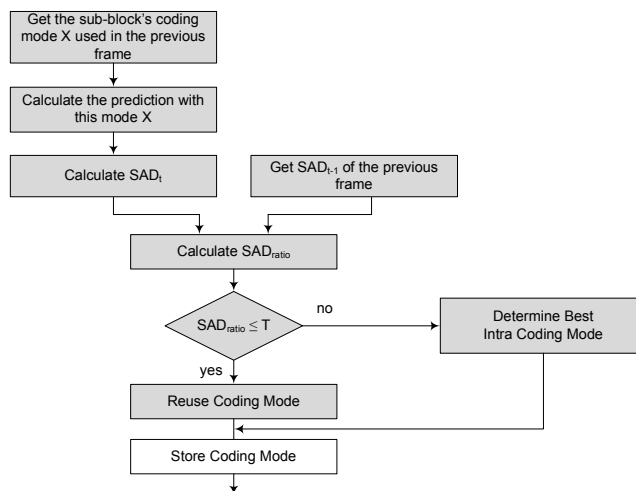


Figure 4.1: Principle work-flow of the proposed single criterion coding mode reuse. This flow-chart shows the coding mode reuse of two co-located sub-blocks depending on the comparison between the SAD_{ratio} and a threshold T .

For the case that the SAD ratio is below a predefined threshold (e.g. 10 percent), the

previously determined coding mode is reused:

$$reuse_mode = \begin{cases} 1 & \text{if } SAD_{ratio} \leq T \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

Otherwise, it is required to determine the correct Intra- 4×4 coding mode by using a full CMS. Figure 4.1 visualizes this decision strategy.

4.1 Impact on Runtime Performance

Before adopting the provided H.264-encoder, a simulation of the proposed method is carried out. This simulation provides information about the improvements on the runtime performance in terms of SAD-costs and mode-reuse. It is therefore necessary to analyse the impact of a varying threshold T and different QPs.

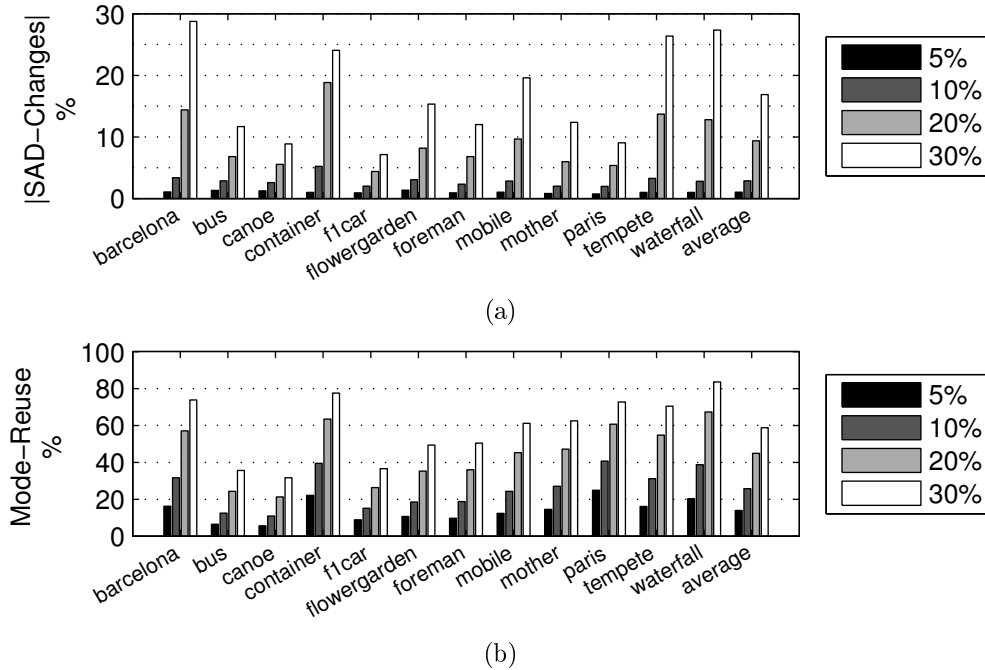


Figure 4.2: Simulation-result of the approach using a single criterion coding mode reuse. This simulation uses the 12 test sequences and a *MATLAB* script for this analysis. Different values for the threshold T (5%, 10%, 20% and 30%) and $QP=30$ are used for the evaluation. (a) SAD-changes related to a full Intra CMS. (b) Reuse of Intra- 4×4 sub-blocks related to the total amount of Intra- 4×4 sub-blocks.

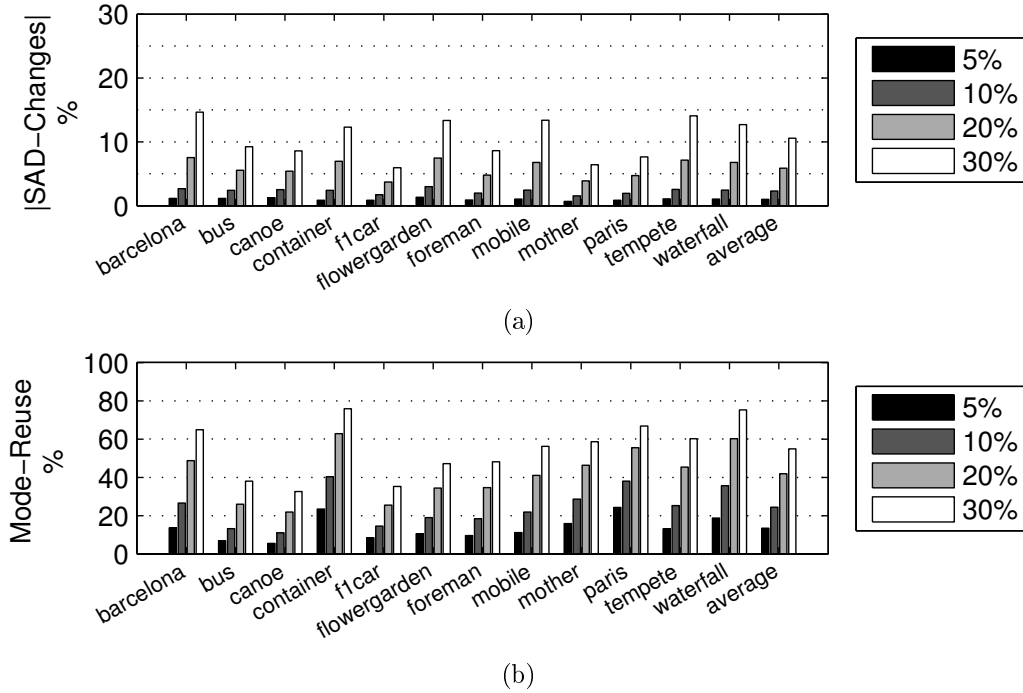


Figure 4.3: Simulation-result of the approach using a single criterion coding mode reuse. This simulation uses the 12 test sequences and a *MATLAB* script for this analysis. Different values for the threshold T (5%, 10%, 20% and 30%) and $QP=40$ are used for the evaluation. (a) SAD-changes related to a full Intra CMS. (b) Reuse of Intra- 4×4 sub-blocks related to the total amount of Intra- 4×4 sub-blocks.

The results are given in Figure 4.2 ($QP=30$) and Figure 4.3 ($QP=40$). These figures are showing the average SAD-changes and the achieved mode-reuse for the given test-sequences.

Considering the SAD-changes (see Figure 4.2(a) and Figure 4.3(a)), there is a noticeable dependency on the adjusted threshold T . The average increases are about 15 percent for a 30 percent threshold.

Next, a relation between the threshold T and the reuse of encoding information (see Figure 4.2(b) and Figure 4.3(b)) can be determined. E.g. a doubling of the threshold also causes a doubling of reused coding modes. The Figures 4.2(b) and 4.3(b) also show, that there is no significant influence of the used QP as the number of reused coding modes is in both cases comparable.

Noticeable variations are only due to the contained motion and texture. This means that sequences having a very slow spatial-motion (e.g. *container*) and / or low texture (e.g. *f1car*) tend to have a higher coding mode reuse compared to high-motion sequences (e.g. *canoe*). The range is between 38% (*canoe*) and 80% (*container*).

4.2 Results

In this section, the achieved performance results of the modified encoder are presented. This evaluation targets the coding mode reuse, the data-rate and the PSNR for different values for the threshold T and various QPs.

4.2.1 Reuse of Coding Modes

The reuse of coding modes is a direct measure for the improvement of the computational complexity. I.e. the higher the number of reused coding modes, the lower is the resulting computational complexity.

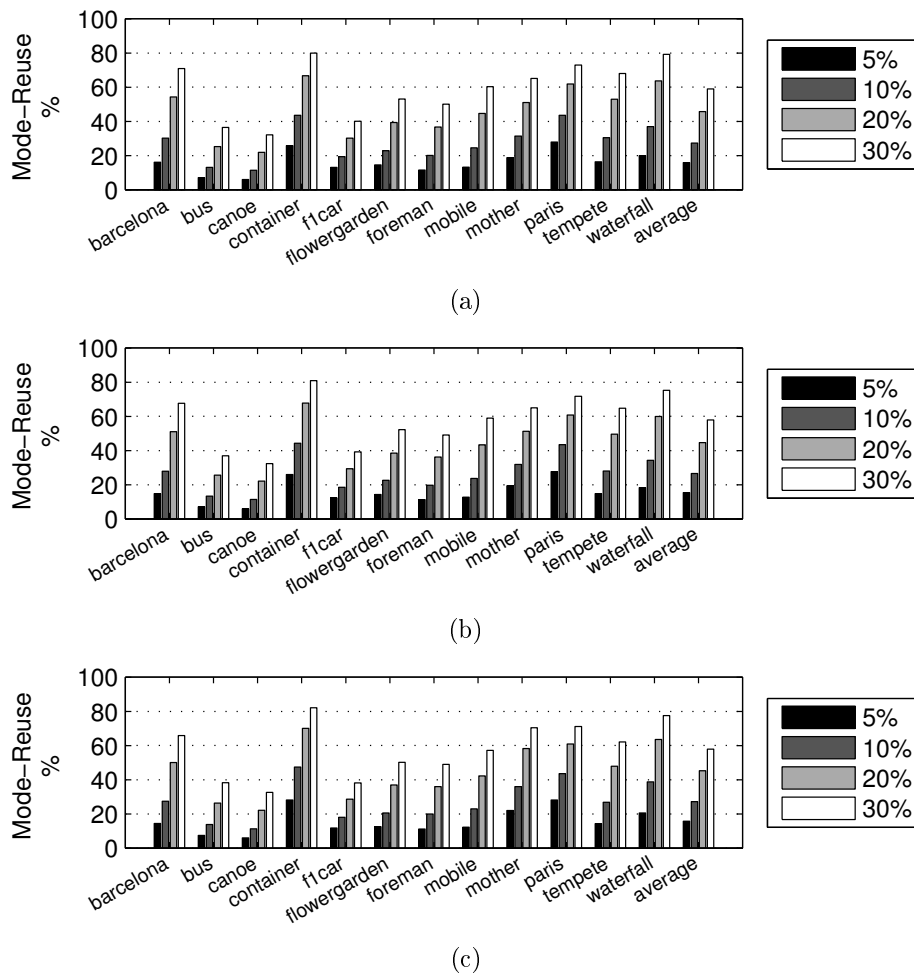


Figure 4.4: Average coding mode reuse of sub-blocks using the single criterion approach. The values are given for each test-sequence and different QPs. (a) Result for QP = 30. (b) Result for QP = 35. (c) Result for QP = 40.

Figure 4.4 shows the evaluated coding mode reuse for a varying threshold T and different QPs. Considering the influence of the QP, it is obvious that there are nearly

no dependencies as the number of reused coding modes is comparable in the Figures 4.4(a), 4.4(b) and 4.4(c).

The average number of sub-blocks with reused coding modes is 16 percent for threshold $T = 5\%$ and reaches almost 60 percent for threshold $T = 30\%$. The variations over the sequences are mostly due to the contained motion. Sequences with a higher amount of motion (e.g. canoe) gain a lower number of reused coding modes.

4.2.2 Coding Efficiency

For evaluating the coding efficiency, in this section the PSNR- and the data-rate changes are analysed. This analysis uses different QPs and threshold values as parameters for examining their influences on the encoding result.

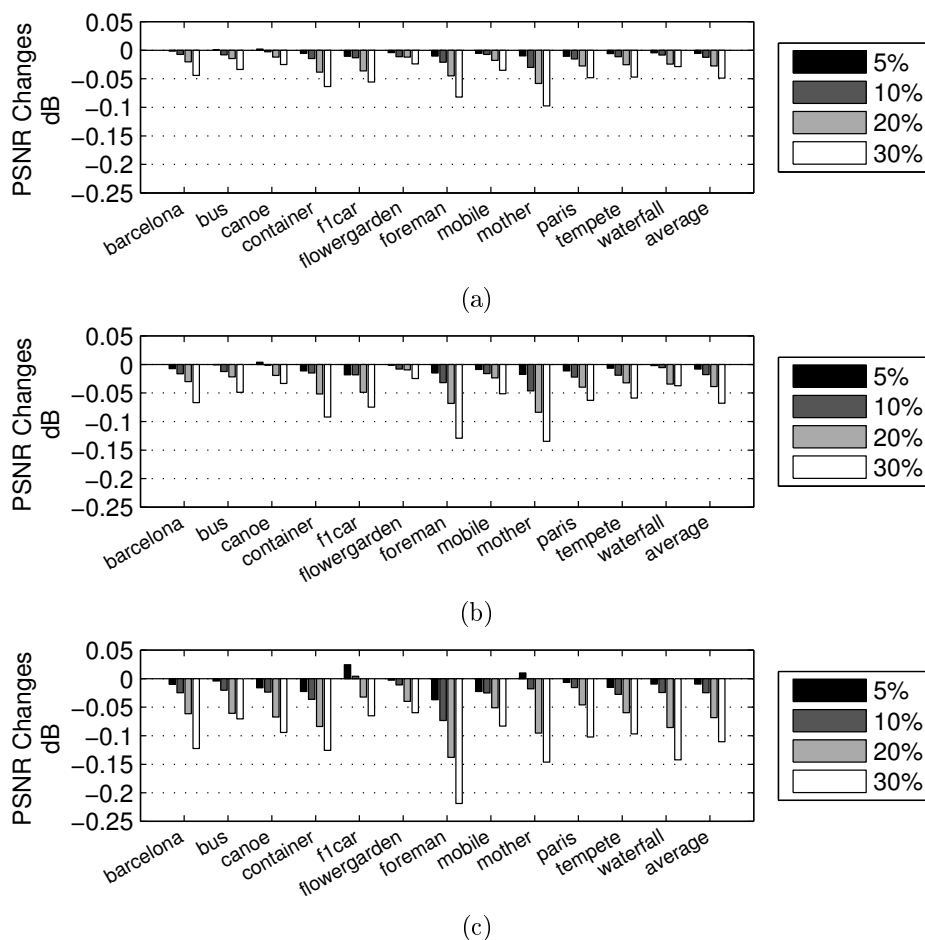
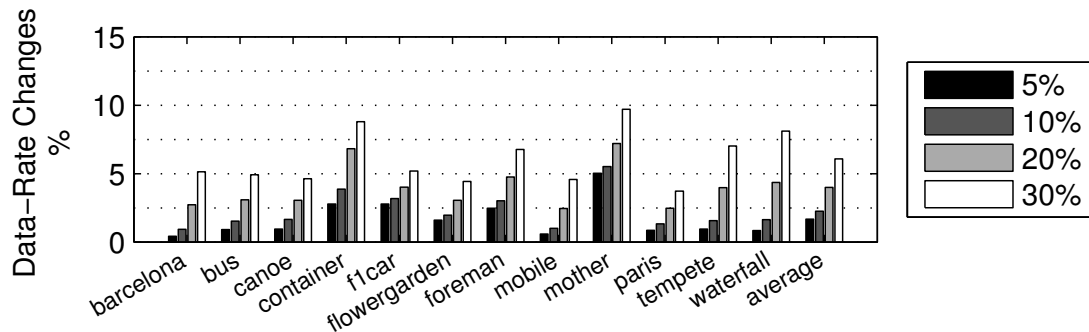
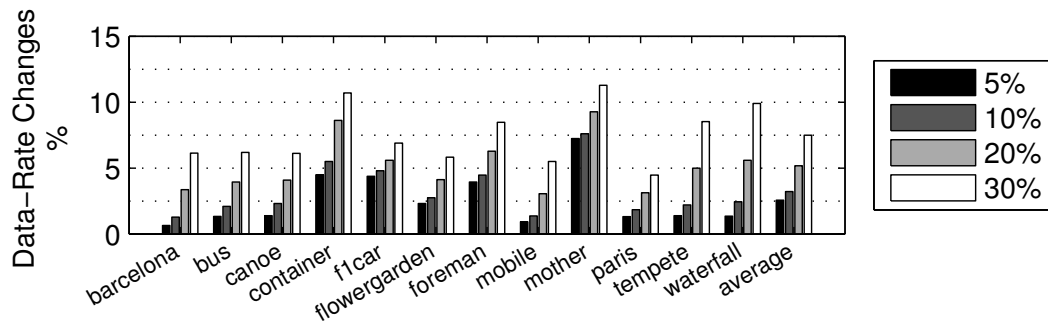


Figure 4.5: Average values of the absolute PSNR differences for the single criterion coding mode reuse. Different values for the threshold T (5%, 10%, 20% and 30%) and QPs are used for the evaluation. (a) Result for QP = 30. (b) Result for QP = 35. (c) Result for QP = 40.

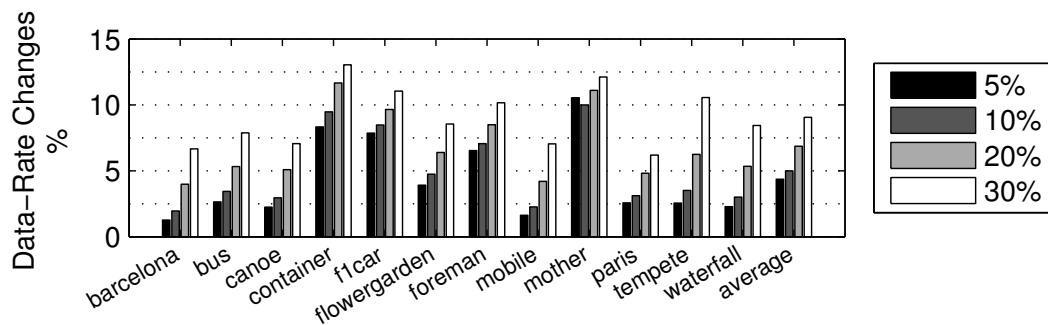
Figure 4.5 shows that the average PSNR-changes are below $0.05dB$ for QP=30 and below $0.10dB$ for QP=40. This means that the visual quality does not suffer from the inaccuracies caused by the mode-reuse.



(a)



(b)



(c)

Figure 4.6: Average data-rate changes for the single criterion coding mode reuse. Different values for the threshold T (5%, 10%, 20% and 30%) and QPs are used for the evaluation. The given values are compared to the achieved data-rates of a full Intra CMS. (a) Result for QP = 30. (b) Result for QP = 35. (c) Result for QP = 40.

For evaluating the changes of the data-rate, Figure 4.6 shows the measured influences when using different threshold values and QPs. Having a 30 percent threshold T , the average increases for $QP=30$ are 4 percent and 8 percent for $QP=40$ respectively.

4.3 Conclusion

Depending on the used parameters (threshold T and QP) the computational complexity of the coding mode selection can be lowered in a range between 15 to 80 percent. However, this lowering leads to a trade-off as the image quality and the data-rates are affected. The visual quality does not change significantly and the changes are most likely not visible for the human eye. It can be noticed that the encoder slightly increased the data-rate for maintaining the visual quality. This low increase of data-rates, while maintaining the PSNR, makes this method applicable for practical applications with low-performance embedded systems.

5 Multi Criteria Coding Mode Reuse (MCCMR)

The promising result of Section 4 showed that the coding mode reuse of static regions lowers the computational complexity of the encoding process. Nevertheless, the drawbacks are small increases of the data-rates and losses of the image-quality (lower PSNR-values). The reason for this behaviour is that the usage of only one discriminating criterion leads to ambiguous results. These uncertainties are causing misclassification of static and non-static sub-blocks.

5.1 Proposed Algorithm

To avoid the above stated prediction failures, in this section a method is presented which incorporates a mixture of spatial and temporal features. The combination of these features is then used as discrimination criterion for the detection of static sub-blocks. Instead of the SCCMR presented in Section 4, a weighted combination of multiple criteria decides on the reuse of a sub-block's coding mode.

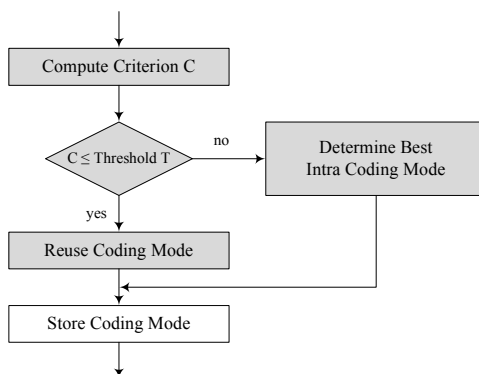


Figure 5.1: The principle work-flow of the proposed MCCMR. This flow-chart shows the coding mode reuse of co-located sub-blocks. The algorithm computes the decision criterion C and compares it to the threshold T .

The proposed decision stage uses multiple different decision criteria (c_0 - c_4). These criteria are explained in detail in the Sections 5.1.1 to 5.1.3. Each criterion c_i is

weighted by a weighting factor w_i and compared to a threshold T :

$$reuse_mode = \begin{cases} 1 & \text{if } \sum_{i=0}^4 c_i w_i \leq T \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

The weighting factors w_i have been determined by observations from the four sequences introduced in Section 3.7.3 (Figure 3.16). As these sequences are different from the 12 test-sequences, an impartial evaluation result of our multi-criterion optimisation technique is possible.

Criterion c_0 exploits the SAD-cost difference between two co-located sub-blocks introduced in Section 4.

5.1.1 Criterion c_1

Criterion c_1 observes the sub-block's coding mode for multiple frames and incorporates long-term knowledge into the proposed algorithm.

$$c_1 = \begin{cases} 1 & \text{if mode constant for } N \text{ frames} \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

If a coding mode is constant for more than N frames, the criterion indicates a highly static region and votes for the reuse of the previous coding mode. Practical experiences showed that $N = 3$ is a proper value and had been used for the following examinations.

5.1.2 Criterion c_2 and c_3

These criteria examine the coding mode reuse of neighbouring sub-blocks. As for a sub-block which is surrounded by sub-blocks with fixed coding modes, it could be assumed that the coding mode of this sub-block is also constant. In larger static regions, these criteria prevent mode changes of single sub-blocks which are surrounded by sub-blocks with reused coding modes. The description of the thereby associated mechanisms are described in Section 3.7.3.

Criterion c_2 is true if the left sub-block reused its previous coding mode:

$$c_2 = \begin{cases} 1 & \text{if left sub-block reused mode} \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

Criterion c_3 uses the upper sub-block instead of the left one. Note that instead of using only the directly neighbouring sub-blocks, these two criteria can propagate knowledge about reused sub-block coding modes to more distant sub-blocks.

5.1.3 Criterion c_4

Criterion c_4 uses the statistical characteristics observed in Figure 3.17 which had been introduced in Section 3.7.3. The first step is to check if the left and upper sub-blocks use the same coding mode and if it is a mode between 0 and 2. In this special case, the coding mode prediction of this current sub-block exploits the heuristic which is given in Figure 3.17. This resulting coding mode is afterwards compared with the mode used in the previous frame. If both modes are equal (i.e. reused mode equals predicted mode) criterion c_4 votes in favour of the coding mode reuse.

5.2 Evaluation of the Modified Encoder

In this section the newly developed algorithm is evaluated by analysing the achieved data-rates, PSNR values and the improvements of the encoder's runtime-behaviour.

5.2.1 Data-Rates

The average data-rate increase is according to Figure 5.2 below three percent. For QP=40 and low textured sequences (e.g. *mother*), strong changes of the data-rates are observed. For these sequences, even small data-rate changes are causing higher relative fluctuations as the data-rates are significantly smaller compared to the other sequences. The sequence *mother*, for instance, takes less than half of the data-rate of sequence *canoe*.

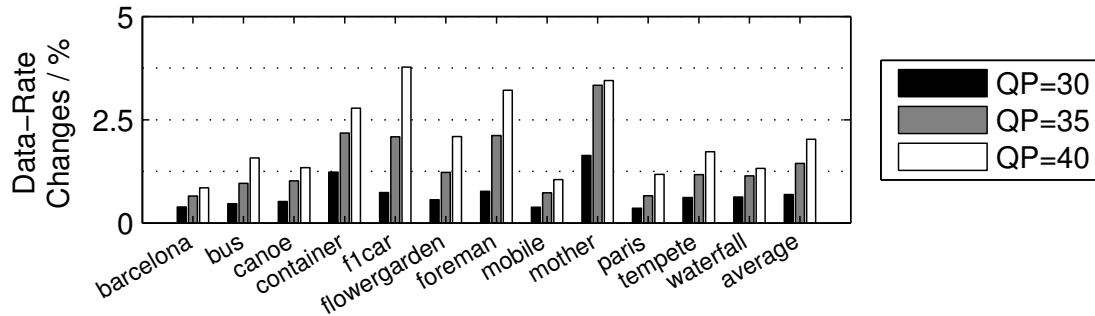


Figure 5.2: Resulting data-rate changes of the implemented multi criterion coding mode reuse. The given values are compared to the full CMS for QP = 30/35/40.

5.2.2 Visual Quality

The test result of the modified H.264 encoder is given in the Figures 5.2, 5.3 and 5.4. These figures are showing the most interesting key facts (PSNR, data-rate and coding mode reuse) for the 12 test-sequences and varying QPs.

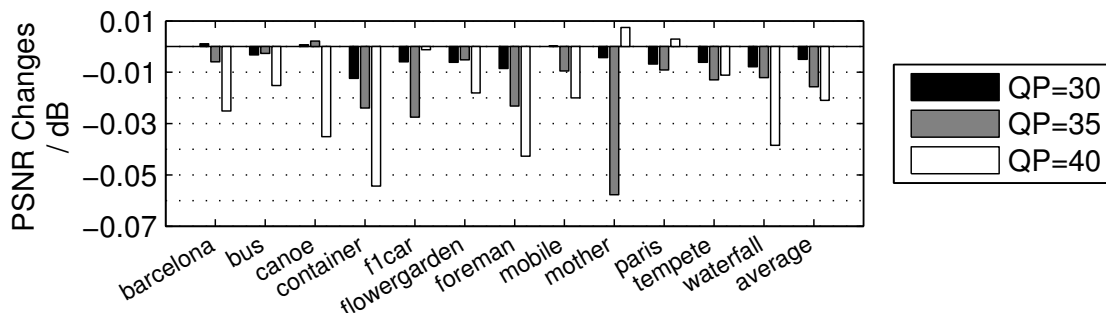


Figure 5.3: Resulting *PSNR* changes of the implemented multi criterion coding mode reuse. The values show the absolute difference between the implemented approach and the full CMS for QP = 30/35/40.

The average PSNR-change is about -0.02dB for QP=40 and below -0.01dB for QP=30 (Figure 5.3). Assumably, these small changes are nearly imperceptible for the human eye.

5.2.3 Runtime Behaviour

Figure 5.4 shows the average coding mode reuse for the 12 test-sequences. Considering the presented bar-graph it is obvious that the coding mode reuse does not significantly depend on the used QP. The average value for QP = 30/35/40 is approximately 30

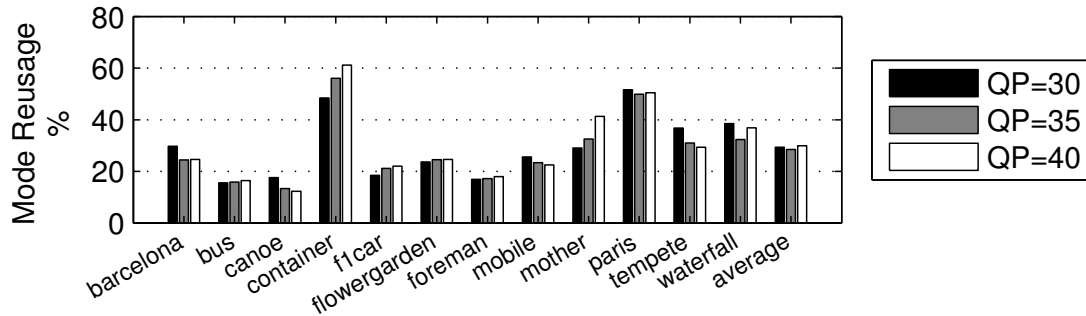


Figure 5.4: Summary of the average coding mode reuse of the implemented multi criterion coding mode reuse. The result shows the evaluation result of the 12 test sequences and $QP = 30/35/40$.

percent. Furthermore, this value is higher for test sequences with little motion (e.g. *container*), as larger parts of the scene are purely static. This indicates that the multi-criterion method works well for different data-rates and texture content.

5.3 Comparison Between SCCMR and MCCMR

A direct comparison of the two developed algorithms is presented in this section. Following to this comparison, the improvements of the runtime performance are shown.

5.3.1 Data-Rate Comparison

The comparison of the relative¹ data-rate changes is given in Figure 5.5. It is shown that the data-rate increases for both methods are similar, but the *multi criterion approach* tends to have better (=lower) data-rates. Furthermore, the resulting data-rates are highly dependent on the used QP and have a fluctuation range of approximately three percent on average.

¹The data-rates are related to the achieved values when using a full CMS.

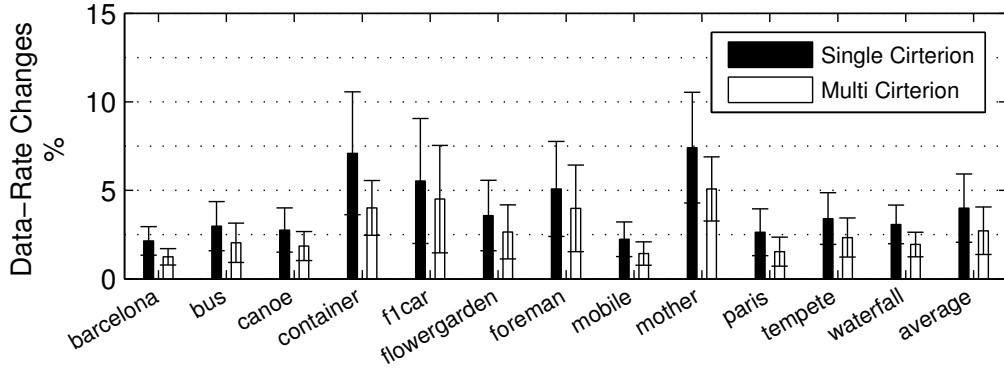


Figure 5.5: Comparison of the data-rate changes between the *single criterion coding mode reuse* and the *multi criterion coding mode reuse*. The bar-graphs are showing the mean values of the achieved data-rates and the value range for $QP = 30/35/40$.

5.3.2 Quality Comparison

This section shows the result of the quality comparison of the SCCMR and the MCCMR. The result is given for various QPs and the 12 test-sequences.

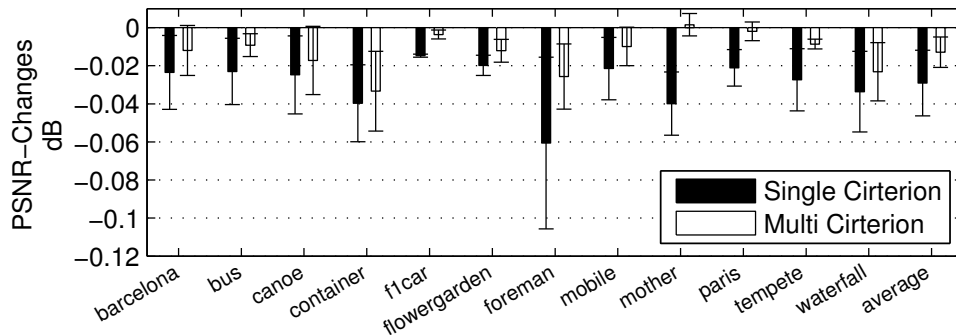


Figure 5.6: Comparison of the average *PSNR*-changes between the *single criterion coding mode reuse* and the *multi criterion coding mode reuse*. The bar-graphs are showing the mean *PSNR*-values and the value range for $QP = 30/35/40$.

Figure 5.6 shows that the *multi criterion approach* gains a higher image quality as the *PSNR* differences are lower. However, both methods are only causing minor changes on the achieved *PSNR*-values. Assumably, these changes are imperceptible for the human eye as they are below 0.03dB on average.

5.3.3 Improvements of the Runtime Behaviour

This section shows the improvements of the runtime behaviour when using the SCCMR or MCCMR.

Figure 5.7 presents the mean values of the achieved Intra- 4×4 coding mode reuse. The illustrated value for each test-sequence is almost constant for QP=30/35/40. The *multi criterion approach* incorporates knowledge about the coding of neighbouring sub-blocks and applies the coding mode reuse less aggressive. Depending on the test-sequence, differences of up to 10 percent are distinguishable (e.g. *barcelona*), however, it is also shown that both methods achieve similar coding mode reuse rates.

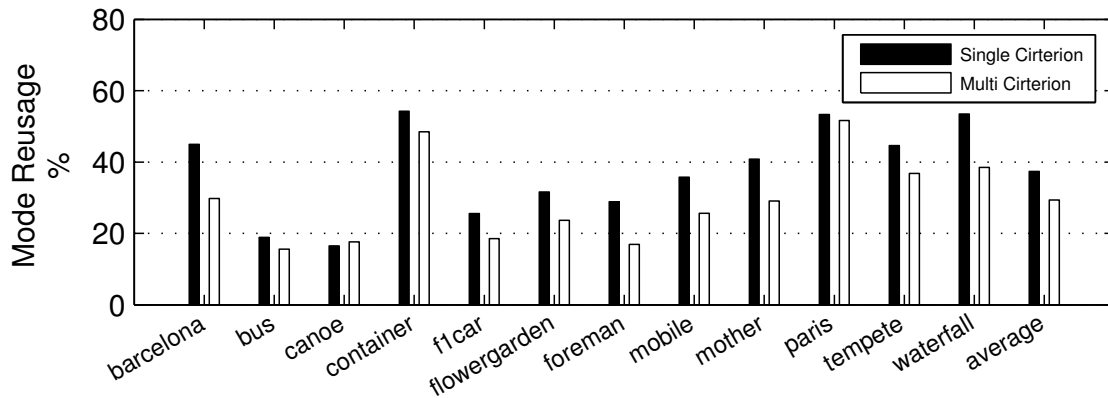


Figure 5.7: Comparison of the coding mode reuse between the two proposed methods. This bar-graph shows the mean values of the achieved results for QP = 30/35/40.

For analysing the computational savings of the modified encoder, it is required to determine the used number of SAD-calculations and compare this value with the number of SAD-calculations which is used for a full Intra- 4×4 CMS. This test used the 12 test-sequences and various quantisation-parameters.

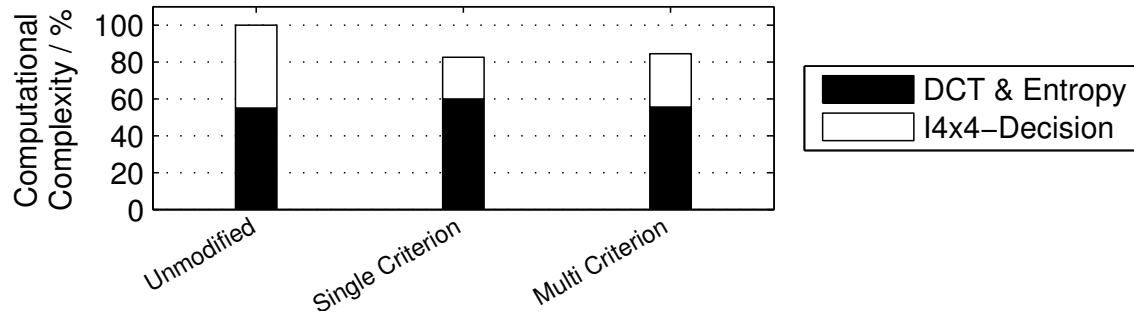


Figure 5.8: Comparison of the runtime complexity between the two proposed algorithms. The figure presents the comparison-result of the runtime between the investigated methods and the unmodified encoder. It is shown that both methods are lowering the computational complexity (approximately 20%) while maintaining the image quality.

Figure 5.8 illustrates the average reduction of the computational complexity of the modified H.264 encoder. Both approaches implement an improved version of the Intra CMS and achieve approximately 20 percent lower encoding times on average.

6 Conclusion

In this thesis, we presented two new algorithms that considerably speed up the runtime of H.264 Intra-only encoders. We focused on speeding up the CMS (coding mode selection) of the spatial encoding that represents the bottleneck in Intra-only encoders. The proposed methods are based on the assumption that for a large number of blocks their spatial prediction modes remain constant over time. Using this assumption our algorithms perform the computationally expensive Intra-4×4 CMS only for a subset of the available 4×4-blocks. For the other 4 × 4-blocks, their coding modes are propagated from the previous video frame. The validity of this assumption has been demonstrated on various test sequences and resulted in a significant reduction of the encoder’s computational complexity.

The first step after setting up this strategy was to determine the impact of using a reduced number of coding modes. We documented that strictly limiting the number of coding modes considerably lowers the visual quality and increases the data-rate of the compressed stream. We concluded that it is necessary to develop algorithms and heuristics for selectively dropping coding modes in order to maintain the coding efficiency. In the following step we therefore developed two methods that skip the computation of coding modes that are unlikely to improve the coding efficiency. The two presented approaches differ in the complexity and accuracy of determining meaningless coding modes for the Intra-4 × 4 CMS. The first method uses a single decision criterion and evaluates the stored information from the previous frame. The second method additionally incorporates further criteria and heuristics to improve the detection of meaningless coding modes.

In our experiments we report that the developed algorithms decrease the computational complexity in the range of 20 to 40 percent, while maintaining similar bitrates and image qualities (PSNR). This result proves the validity of the chosen assumptions and strategies for speeding up the runtime of H.264 Intra-only encoders.

In future work, the proposed algorithms could be adapted by using more decision criteria to further improve the detection of meaningless coding modes. It is also recommended to include strategies for handling 16 × 16 large macroblocks. This measure is intended to lower the bitrates of the compressed stream.

Glossary and Acronyms

4:2:0 In 4:2:0 the chrominance components Cb and Cr are sub-sampled at a factor of 2 in horizontal and vertical direction.

AVC Advanced Video Coding. See H.264.

CIF Common Intermediate Format: Is a format used to standardize the horizontal and vertical resolutions in pixels of different video signals (352x288 Pixels).

CODEC enCoder / DECoder

DCT Discrete Cosine Transformation. A DCT expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations.

DVD Digital Versatile Disc

GCC GNU Compiler Collection: The GNU Compiler Collection is a compiler system supporting various programming languages produced by the GNU Project.

H.262 Is a standard for video compression and is equivalent to MPEG-2. It had been introduced in 1994 and is currently used by several technologies like DVD and so on.

H.263 Is a standard for video compression. It had been introduced in 1995.

H.264 Is a standard for video compression and is equivalent to MPEG-4 Part 10.

JVT Joint Video Team (JVT). The Joint Video Team is a group of video coding experts from the ITU-T.

MAD Mean of Absolute Differences: It works by taking the mean value of the difference between each pixel in the original block and the corresponding pixel in the block being used for comparison.

MATLAB Is a numerical computing environment and programming language. This software is intended for scientists and mathematicians for solving numerical problems.

MB macroblock

MPEG ISO Motion Picture Experts Group (MPEG). It was formed by the ISO to set standards for audio and video compression and transmission.

MSE Mean Squared Error: The mean squared error is a possibility to quantify the amount by which an estimator differs from the original value.

MV motion-vector

NAL Network Abstraction Layer (NAL).

ODM ON DEMAND Microelectronics AG.

PSNR Peak Signal to Noise Ratio: Measures the image quality of both still images and motion images.

QP Quantisation Parameter.

SAD Sum of Absolute Differences: It works by taking the absolute value of the difference between each pixel in the original block and the corresponding pixel in the block being used for comparison. These differences are summed to create a simple metric of block similarity.

VCEG ITU-T Video Coding Experts Group (VCEG) is the informal name for Video coding of the ITU-T. It is responsible for standardization of the *H.26x* line of video coding standards and related technologies.

A Appendix

A.1 Prediction Table

Mode left	Mode upper	Most probable modes
0	0	0, 2, 1, 5, 7, 3, 4, 8, 6
0	1	0, 1, 2, 8, 6, 5, 3, 7, 4
0	2	0, 2, 1, 5, 8, 7, 3, 4, 6
0	3	0, 2, 1, 7, 8, 5, 3, 4, 6
0	4	0, 2, 1, 5, 6, 8, 4, 7, 3
0	5	0, 2, 5, 1, 7, 4, 8, 3, 6
0	6	0, 1, 2, 6, 8, 5, 4, 3, 7
0	7	0, 2, 7, 1, 5, 3, 8, 4, 6
0	8	0, 1, 2, 8, 6, 7, 5, 3, 4
1	0	1, 0, 2, 8, 6, 4, 5, 7, 3
1	1	1, 8, 2, 6, 0, 4, 3, 7, 5
1	2	1, 2, 8, 6, 0, 4, 3, 7, 5
1	3	1, 2, 8, 6, 0, 3, 4, 5, 7
1	4	1, 8, 2, 6, 0, 4, 3, 5, 7
1	5	1, 8, 2, 6, 0, 4, 3, 5, 7
1	6	1, 8, 6, 2, 0, 4, 3, 5, 7
1	7	1, 8, 2, 0, 6, 7, 3, 4, 5
1	8	1, 8, 2, 6, 0, 3, 4, 7, 5
2	0	2, 0, 1, 5, 6, 8, 4, 7, 3
2	1	1, 2, 8, 6, 0, 4, 3, 5, 7
2	2	2, 1, 0, 6, 8, 4, 3, 5, 7
2	3	2, 1, 0, 8, 6, 3, 4, 5, 7
2	4	2, 1, 0, 6, 8, 4, 5, 3, 7
2	5	2, 0, 1, 5, 6, 8, 4, 3, 7
2	6	1, 2, 6, 8, 0, 4, 3, 5, 7
2	7	2, 0, 1, 8, 3, 5, 7, 6, 4
2	8	1, 2, 8, 6, 0, 3, 4, 5, 7

3	0	0, 2, 1, 8, 3, 7, 6, 5, 4
3	1	1, 2, 8, 3, 6, 0, 7, 4, 5
3	2	2, 1, 3, 8, 0, 6, 4, 7, 5
3	3	2, 3, 8, 1, 0, 7, 6, 4, 5
3	4	2, 1, 8, 3, 6, 0, 4, 7, 5
3	5	2, 0, 8, 3, 1, 5, 6, 7, 4
3	6	1, 2, 8, 6, 3, 0, 4, 5, 7
3	7	2, 3, 8, 1, 0, 7, 5, 6, 4
3	8	8, 1, 2, 3, 6, 7, 0, 4, 5
4	0	0, 1, 2, 6, 5, 8, 4, 7, 3
4	1	1, 2, 8, 6, 0, 4, 3, 5, 7
4	2	2, 1, 8, 0, 6, 4, 5, 3, 7
4	3	1, 2, 8, 6, 0, 4, 5, 3, 7
4	4	6, 1, 4, 2, 0, 8, 5, 3, 7
4	5	0, 5, 1, 2, 4, 6, 8, 7, 3
4	6	1, 6, 2, 0, 8, 4, 5, 3, 7
4	7	0, 2, 1, 8, 6, 4, 5, 3, 7
4	8	1, 8, 2, 6, 0, 4, 3, 5, 7
5	0	0, 5, 2, 1, 7, 3, 8, 4, 6
5	1	1, 0, 2, 8, 6, 5, 4, 3, 7
5	2	0, 2, 1, 5, 8, 6, 4, 7, 3
5	3	0, 5, 2, 1, 8, 6, 7, 4, 3
5	4	0, 5, 2, 1, 4, 6, 7, 8, 3
5	5	5, 0, 2, 4, 1, 7, 8, 6, 3
5	6	1, 0, 2, 5, 6, 8, 4, 7, 3
5	7	0, 5, 2, 1, 7, 8, 3, 4, 6
5	8	8, 1, 0, 2, 5, 6, 4, 7, 3
6	0	0, 1, 6, 2, 8, 4, 5, 3, 7
6	1	1, 6, 8, 2, 0, 4, 3, 5, 7
6	2	1, 6, 2, 8, 0, 4, 3, 5, 7
6	3	1, 6, 8, 2, 0, 4, 3, 7, 5
6	4	1, 6, 8, 2, 4, 0, 3, 5, 7
6	5	1, 6, 2, 8, 0, 4, 5, 3, 7
6	6	6, 1, 8, 2, 0, 4, 3, 5, 7
6	7	1, 8, 6, 2, 0, 4, 3, 5, 7
6	8	1, 8, 6, 2, 0, 4, 3, 5, 7

7	0	0, 2, 7, 1, 5, 3, 8, 4, 6
7	1	1, 0, 2, 8, 7, 6, 3, 4, 5
7	2	2, 0, 7, 1, 3, 6, 8, 5, 4
7	3	7, 0, 2, 1, 3, 8, 6, 5, 4
7	4	0, 7, 2, 1, 3, 5, 8, 4, 6
7	5	0, 7, 2, 3, 1, 5, 8, 6, 4
7	6	1, 0, 2, 3, 7, 8, 6, 5, 4
7	7	7, 0, 2, 1, 3, 8, 5, 6, 4
7	8	8, 1, 0, 7, 2, 6, 3, 4, 5
8	0	1, 0, 8, 2, 6, 3, 7, 4, 5
8	1	1, 8, 2, 6, 0, 3, 4, 7, 5
8	2	1, 8, 2, 6, 0, 3, 4, 7, 5
8	3	1, 8, 2, 6, 3, 0, 4, 7, 5
8	4	1, 8, 2, 6, 0, 3, 4, 5, 7
8	5	1, 8, 2, 0, 6, 3, 4, 5, 7
8	6	1, 8, 6, 2, 0, 3, 4, 5, 7
8	7	1, 8, 2, 0, 3, 6, 7, 4, 5
8	8	1, 8, 2, 6, 3, 0, 4, 7, 5

Bibliography

- [1] Z. Wei, H. Li, and K. N. Ngan, "An efficient intra mode selection algorithm for H.264 based on fast edge classification," *IEEE International Symposium on Circuits and Systems ISCAS*, pp. 3630 – 3633, June 2007.
- [2] H. Li and K. N. Ngan, "Fast and efficient method for block edge classification," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, (New York), pp. 67 – 72, ACM, 2006.
- [3] C. L. Yang, L. M. Po, and W. H. Lam, "A fast H.264 intra prediction algorithm using macroblock properties," *International Conference on Image Processing (ICIP)*, vol. 1, pp. 461 – 464, 2004.
- [4] C. Hwang, S. S. Zhuang, and S. H. Lai, "Efficient intra mode selection using image structure tensor for H.264 / AVC," *IEEE International Conference on Image Processing (ICIP)*, vol. 5, pp. 289 – 292, September 2007.
- [5] U. Mithun, "An early intra mode skipping technique for inter frame coding in H.264 baseline-profile," *Digest of Technical Papers for the International Conference on Consumer Electronics*, pp. 1 – 2, 2007.
- [6] B. Meng, O. C. Au, C. W. Wong, and H. K. Lam, "Efficient intra-prediction mode selection for 4x4 blocks in H.264," *In Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 3, pp. 521 – 524, July 2003.
- [7] J. Kim and J. Jeong, "Fast intra-mode decision in H.264 video coding using simple directional masks," *International Conference on Visual Communications and Image Processing*, vol. 5960, pp. 1 – 10, July 2005.
- [8] J. S. Ryu and E. T. Kim, "Fast intra coding method of H.264 for video surveillance system," *IJCSNS International Journal of Computer Science and Network Security*, vol. 7, no. 10, pp. 76 – 81, 2007.
- [9] A. C. Tsai, A. Paul, J. C. Wang, and J. F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 694 – 698, May 2008.

- [10] W. Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs," *Journal of The American Society for Information Science.*, vol. 49, pp. 633 – 648, 1998.
- [11] M. C. Hwang, J. K. Cho, J. S. Kim, J. H. Kim, and S. J. Ko, "Fast intra prediction mode selection scheme using temporal correlation in H.264," *IEEE Conference on Innovative Technologies for Societal Transformation*, pp. 1 – 5, November 2005.
- [12] H. Ko, K. Yoo, J. Seo, and K. Sohn, "Fast intra-mode decision using inter-frame correlation from H.264/AVC," *IEEE International Symposium on Consumer Electronics*, pp. 1 – 4, April 2008.
- [13] J. Xin and A. Vetro, "Fast mode decision for intra-only H.264/AVC coding," *Proceedings of the 25th Picture Coding Symposium*, pp. 7 – 14, April 2006.
- [14] C. C. Wang, T. S. Chen, and C. W. Tung, "Fast intra-mode decision in H.264 using inter-block correlation," *International Conference on Image Processing (ICIP)*, pp. 1345 – 1348, October 2006.
- [15] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd, 2003.
- [16] G. Sullivan and T. Wiegand, "Video compression: From concepts to the H.264 / AVC standard," *Proceedings of the IEEE*, vol. 93, pp. 18 – 31, January 2005.
- [17] I. E. Richardson, "H.264 / MPEG-4 part 10 transform and quantisation." White Paper, March 2003. <http://www.vcodex.com>.
- [18] G. Hwang, J. Park, B. Jung, K. Choi, Y. Joo, Y. Oh, and B. Jeon, "Efficient fast intra mode decision using transform coefficients," *The 9th International Conference on Advanced Communication Technology*, pp. 399 – 402, February 2007.
- [19] T. Tsukuba, I. Nagayoshi, T. Hanamura, and H. Tominaga, "H.264 fast intra-prediction mode decision based on frequency characteristic," *Proceedings of EUSIPCO2005*, pp. 61 – 66, September 2005.
- [20] M. Jafari and S. Kasaei, "Fast intra- and inter-prediction mode decision in H.264 advanced video coding," *IEEE International Journal of Computer Science and Network Security*, vol. 8, pp. 1 – 6, May 2008.
- [21] J. Fritts, H. Zhang, and F. Steiling, "Fast intra-prediction mode selection for H.264," *SPIE05*, p. 23, 2005.

- [22] A. Elyousfi, A. Tamtaoui, and E. Bouyakhf, "A new fast intra prediction mode decision algorithm for H.264/AVC encoders," *International Journal of Computer Systems Science and Engineering*, vol. 4, pp. 1 – 7, December 2006.
- [23] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 793 – 849, 2004.
- [24] "ITU-T Recommendation H.264: Advanced Video Coding for Generic Audiovisual Services," November 2007. <http://www.itu.int/rec/T-REC-H.264-200711-I/en>.
- [25] F. Seitner, J. Meser, G. Schedelberger, A. Wasserbauer, M. Bleyer, M. Gelautz, M. Schutti, R. Schreier, P. Vaclavik, G. Krottendorfer, G. Truhlar, T. Bauernfeind, and P. Beham, "Design Methodology for the SVENm Multimedia Engine," *Austrochip*, 2008.
- [26] T. Mathworks. <http://www.mathworks.com>, March 2009.
- [27] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," *ITU SG16 VCEG-M33*, April 2001.

List of Figures

1.1	Coding mode selection using temporal correlation.	3
2.1	Block-diagram of an H.264 encoder. The gray shaded blocks are representing an encoder only using spatial prediction methods. The white blocks are used for encoders supporting temporal prediction too.	7
2.2	Block-diagram of an H.264 decoder. The gray shaded blocks are representing a decoder only using spatial prediction methods. The white blocks are used for decoders supporting temporal decoding.	8
2.3	H.264 Baseline, Main and Extended profiles.	9
2.4	Structuring of H.264 into <i>Frames</i> , <i>Slices</i> and <i>MBs</i>	10
2.5	Neighbours used for spatial prediction of the current <i>MB</i>	10
2.6	Intra 16×16 luminance prediction modes.	11
2.7	Partitioning schema of a 16×16 luma <i>MB</i>	12
2.8	Labelling of prediction samples for a 4×4 sub-block.	12
2.9	Intra 4×4 luminance prediction modes.	13
2.10	Intra 4×4 luminance prediction mode examples.	14
2.11	4:2:0 chroma sub-sampling.	15
2.12	Intra- 8×8 chrominance prediction modes.	16
2.13	Principle work-flow of the used H.264 Intra-mode selection.	17
2.14	Runtime profiling of the encoding process for the 12 test sequences.	18
3.1	Overview of the used test sequences.	20
3.2	Determining the MV by minimising the SAD of corresponding MBs within a given search-area (40×40 pixels).	21
3.3	Determining the norm of the limiting MVs.	22
3.4	Data-rate changes for Intra-only encoding using either 4×4 or 16×16 coding modes.	27
3.5	PSNR changes for Intra-only encoding using either 4×4 or 16×16 coding modes.	29
3.6	Summary of the PSNR changes for Intra-only encoding using either 4×4 or 16×16 coding modes.	29
3.7	Average frequency of the coding modes.	30
3.8	Relative Data-Rate changes for a reduced number of Intra coding modes.	31

3.9	PSNR changes for a reduced number of Intra coding modes.	32
3.10	Determining the magnitude of the limiting Motion Vector.	34
3.11	Interrelation of the mode retention and image motion for the 12 test sequences and $QP = 30$	35
3.12	Interrelation of the mode retention and image motion for the 12 test sequences and $QP = 35$	35
3.13	Interrelation of the mode retention and image motion for the 12 test sequences and $QP = 40$	35
3.14	SAD-cost changes of static sub-blocks with alternating coding modes.	36
3.15	SAD-cost changes of static sub-blocks with unchanged coding mode.	37
3.16	Four extra sequences for analysing assumption $A3$	38
3.17	Matrix showing the prediction-heuristic. The values in the array are the most likely modes and the probability values.	39
3.18	Probability of the coding mode reuse for sub-blocks where the left and/or upper neighbours keep their coding mode.	40
4.1	Principle work-flow of the proposed single criterion coding mode reuse.	42
4.2	Simulation-result of the approach using a single criterion coding mode reuse ($QP=30$).	43
4.3	Simulation-result of the approach using a single criterion coding mode reuse ($QP=40$).	44
4.4	Average coding mode reuse of sub-blocks using the single criterion approach.	45
4.5	Average values of the absolute PSNR differences for the single criterion coding mode reuse.	46
4.6	Average data-rate changes for the single criterion coding mode reuse.	47
5.1	The principle work-flow of the proposed MCCMR.	49
5.2	Average data-rate changes of the implemented multi criterion coding mode reuse.	52
5.3	Average <i>PSNR</i> changes of the implemented multi criterion coding mode reuse.	52
5.4	Summary of the average coding mode reuse of the implemented multi criterion coding mode reuse.	53
5.5	Comparison of the data-rate changes between the <i>single criterion coding mode reuse</i> and the <i>multi criterion coding mode reuse</i>	54
5.6	Comparison of the average <i>PSNR</i> -changes between the <i>single criterion coding mode reuse</i> and the <i>multi criterion coding mode reuse</i>	54

5.7	Comparison of the coding mode reuse between the proposed approaches.	55
5.8	Comparison of the runtime complexity between the two proposed algorithms.	56

List of Tables

2.1	Overview of the supported Intra coding modes.	16
3.1	Definition of the classification-groups used for the MV-distribution.	22
3.2	Classified MVs for the test sequences barcelona, bus, canoe, container, f1car and flowergarden.	24
3.3	Classified MVs for the test sequences foreman, mobile, mother, paris, tempete and waterfall.	25