Fast Mode Selection Algorithm for H.264 Video Coding

Examensarbete utfört i Bildkodning
vid Linköpings tekniska högskola
av

Ola Hållmarker
Martin Linderoth

LITH-ISY-EX–05/3684–SE
Linköping 2005
Fast Mode Selection Algorithm for H.264 Video Coding

Examensarbete utfört i Bildkodning
vid Linköpings tekniska högskola
av

Ola Hållmarker
Martin Linderoth

LITH-ISY-EX–05/3684–SE

Handledare: Magnus Hoem, Popwire Technology
Examinator: Robert Forchheimer
Linköping 2005
Algoritm för effektivt val av mod för H.264 videokodning

Fast Mode Selection Algorithm for H.264 Video Coding

Ola Hållmarker, Martin Linderoth

Abstract

ITU-T and the Moving Picture Expert Group (MPEG) have jointly, under the name of Joint Video Team (JVT), developed a new video coding standard. The standard is called H.264 and is also known as Advanced Video Coding (AVC) or MPEG-4 part 10. Comparisons shows that H.264 greatly outperforms MPEG-2, currently used in DVD and digital TV. H.264 halves the bit rate with equal image quality. The great rate-distortion performance means nevertheless a high computational complexity. Especially on the encoder side.

Handling of audio and video, e.g. compressing and filtering, is quite complex and requires high performance hardware and software. A video encoder consists of a number of modules that find the best coding parameters. For each macroblock several modes are evaluated in order to achieve optimal coding. The reference implementation of H.264 uses a brute force search for this mode selection which is extremely computational constraining. In order to perform video encoding with satisfactory speed there is an obvious need for reducing the amount of modes that are evaluated.

This thesis proposes an algorithm which reduces the number of modes and reference frames that are evaluated. The algorithm can be regulated in order to fulfill the demand on quality versus speed. Six times faster encoding can be obtained without loosing perceptual image quality. By allowing some quality degradation the encoding becomes up to 20 times faster.

Nyckelord

Advanced Video Coding, AVC, H.264, Mode Selection, MPEG-4 Part 10, Multiple Reference Frames, Real Time Coding.
Abstract

ITU - T and the Moving Picture Expert Group (MPEG) have jointly, under the name of Joint Video Team (JVT), developed a new video coding standard. The standard is called H.264 and is also known as Advanced Video Coding (AVC) or MPEG-4 part 10. Comparisons shows that H.264 greatly outperforms MPEG-2, currently used in DVD and digital TV. H.264 halves the bit rate with equal image quality. The great rate - distortion performance means nevertheless a high computational complexity. Especially on the encoder side.

Handling of audio and video, e.g. compressing and filtering, is quite complex and requires high performance hardware and software. A video encoder consists of a number of modules that find the best coding parameters. For each macroblock several modes are evaluated in order to achieve optimal coding. The reference implementation of H.264 uses a brute force search for this mode selection which is extremely computational constraining. In order to perform video encoding with satisfactory speed there is an obvious need for reducing the amount of modes that are evaluated.

This thesis proposes an algorithm which reduces the number of modes and reference frames that are evaluated. The algorithm can be regulated in order to fulfill the demand on quality versus speed. Six times faster encoding can be obtained without loosing perceptual image quality. By allowing some quality degradation the encoding becomes up to 20 times faster.

Keywords: Advanced Video Coding, AVC, H.264, Mode Selection, MPEG-4 Part 10, Multiple Reference Frames, Real Time Coding.
Acknowledgements

We would like to thank all the nice people at Popwire Technology\textsuperscript{1}, especially our supervisors Magnus Hoem and Pontus Carlsson, for supporting us in our work. We would also like to thank our examiner Robert Forchheimer and our opponents Daniel Bernardsson and Johan Törne for their helpful ideas and accurate correction of this report.

Martin Linderoth and Ola Hållmarker
April 2005

\textsuperscript{1}www.popwire.com
Contents

1 Introduction 1
   1.1 Purpose .................................................. 1
   1.2 Project Review .......................................... 1
   1.3 Report Outline ......................................... 2

2 Video Coding in General 3
   2.1 Introduction .............................................. 3
   2.2 Color Spaces ............................................. 4
      2.2.1 RGB .................................................. 4
      2.2.2 YCbCr ............................................... 4
   2.3 Interlaced video ......................................... 5
   2.4 Quality .................................................. 6
      2.4.1 Objective Quality Measures ....................... 6
      2.4.2 Subjective Quality Measures ..................... 6
   2.5 Codec Overview ......................................... 7
      2.5.1 Encoder .............................................. 7
      2.5.2 Decoder ............................................. 7
   2.6 Predictive Coding ....................................... 8
   2.7 Motion Estimation and Compensation ................... 9
   2.8 Transform and Quantization ............................. 13
      2.8.1 Transform Coding .................................. 13
      2.8.2 Quantization ....................................... 14
   2.9 Entropy Coding ......................................... 15
      2.9.1 Huffman Coding .................................... 15
      2.9.2 Arithmetic Coding .................................. 16
3 H.264
3.1 Structure ........................................... 19
3.2 Profiles ............................................ 22
3.3 Intra Coding ....................................... 22
3.4 Motion Estimation and Compensation ............... 25
   3.4.1 Multiple reference frames .................... 25
   3.4.2 Block Partitioning ............................ 25
   3.4.3 Subpixel Motionsearch ....................... 26
3.5 Deblocking Filter .................................. 26
3.6 Transform and Quantization ........................ 27
   3.6.1 Transform Coding ............................ 27
   3.6.2 Quantization ................................. 27
3.7 Entropy Coding .................................... 28

4 Statistical Analysis ................................. 29
4.1 Introduction ....................................... 29
4.2 Mode Statistics .................................... 29
   4.2.1 Selection of Best Mode ....................... 30
   4.2.2 Influence of Mode in Earlier Macroblocks ... 32
   4.2.3 Reference Frames ............................. 34
4.3 Macroblock Measures ................................ 36
   4.3.1 Energy Estimating Measurers ................. 36
   4.3.2 Rate Distortion Cost ......................... 41
4.4 List of Observations ................................ 43

5 Optimization of Mode Selection ..................... 45
5.1 Introduction ....................................... 45
5.2 Intra Mode Predictors ................................ 45
   5.2.1 Intra 16x16 or Intra 4x4 Predictor ........... 46
   5.2.2 Intra 4x4 Predictor ........................... 48
   5.2.3 Combined Intra Predictor ..................... 49
   5.2.4 Results ....................................... 49
5.3 Inter Mode Predictors .............................. 52
   5.3.1 Preprocessing ................................ 52
   5.3.2 Skip Predictor ............................... 52
   5.3.3 Skip and Mode 16x16 Predictor ............... 57
   5.3.4 Mode 8x8 Predictor ........................... 60
   5.3.5 Submode Predictor ............................ 63
## CONTENTS

5.3.6 Rate Distortion Cost Predictor ........................................ 66
5.3.7 Results ........................................................................... 66
5.4 Reference Frame Predictor .................................................. 70
5.5 Scene Change Detection ...................................................... 73
5.5.1 Results ........................................................................... 73
5.6 Regulator ........................................................................... 75
5.6.1 Results ........................................................................... 76
5.7 Mode Selection Predictor .................................................... 77
5.7.1 Results ........................................................................... 80

6 Proposed Algorithm ............................................................... 83
6.1 Results .............................................................................. 84
6.2 Conclusions/Discussion ........................................................ 89

7 Future Work .......................................................................... 91

A Mathematical Formulas .......................................................... 99

B Calculation of PSNR Drop and Increase of Rate ......................... 103
# List of Figures

2.1 Sampling formats ............................................. 5  
2.2 Overview of a codec ........................................... 7  
2.3 Overview of an encoder ....................................... 8  
2.4 Overview of a decoder ....................................... 9  
2.5 Residual ..................................................... 10  
2.6 Motion compensated residual ............................... 11  
2.7 Linear quantizer ............................................. 14  
2.8 Huffman tree ................................................ 16  
2.9 Arithmetic coding - Probability intervals for the sym-  
bol ................................................................. 18  
3.1 QCIF Frame consisting of two slices ....................... 19  
3.2 Macroblock and subblock partitions ......................... 20  
3.3 Macroblock and subblock partitions ......................... 21  
3.4 Prediction modes for intra 16x16 prediction ............... 23  
3.5 Prediction modes for intra 4x4 prediction ................ 24  
3.6 Subpel interpolation ........................................... 27  
4.1 Probability of optimal modes for Claire .................. 31  
4.2 Probability of optimal modes for Foreman ................. 32  
4.3 Temporal correlation for Foreman, QP 24 .................. 33  
4.4 Spatial correlation for Carphone, QP 32 .................... 34  
4.5 Reference frames ............................................. 35  
4.6 SAD distribution for Foreman 40 ............................. 37  
4.7 Pixdiff distribution for Foreman 40 ......................... 39  
4.8 Mean pixdiff for various QP ................................. 39  
4.9 Pixdiff in a macroblock for Intra 16x16 and Intra 4x4 in  
Scenecut, QP 40 ................................................. 40
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Overview of the intra 16x16 or intra 4x4 predictor</td>
<td>47</td>
</tr>
<tr>
<td>5.2</td>
<td>Overview of Intra 4x4 predictor</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>Overview of skip predictor</td>
<td>54</td>
</tr>
<tr>
<td>5.4</td>
<td>Rate Distortion curve for skip predictor for Carphone.</td>
<td>56</td>
</tr>
<tr>
<td>5.5</td>
<td>Overview of skip and mode 16x16 predictor</td>
<td>57</td>
</tr>
<tr>
<td>5.6</td>
<td>Rate Distortion curve for skip or mode 16x16 predictor for Coastguard</td>
<td>58</td>
</tr>
<tr>
<td>5.7</td>
<td>Overview of mode 8x8 predictor</td>
<td>60</td>
</tr>
<tr>
<td>5.8</td>
<td>Rate Distortion curve for mode 8x8 predictor for Foreman.</td>
<td>61</td>
</tr>
<tr>
<td>5.9</td>
<td>Overview of submode predictor</td>
<td>63</td>
</tr>
<tr>
<td>5.10</td>
<td>Rate Distortion curve for subblock predictor for Carphone.</td>
<td>64</td>
</tr>
<tr>
<td>5.11</td>
<td>Overview of the RD cost predictor</td>
<td>67</td>
</tr>
<tr>
<td>5.12</td>
<td>Rate Distortion curve for RD cost predictor for Foreman.</td>
<td>69</td>
</tr>
<tr>
<td>5.13</td>
<td>Overview of reference predictor</td>
<td>71</td>
</tr>
<tr>
<td>5.14</td>
<td>Overview of the scene change predictor</td>
<td>74</td>
</tr>
<tr>
<td>5.15</td>
<td>Overview of the regulator</td>
<td>75</td>
</tr>
<tr>
<td>5.16</td>
<td>Results of the regulator</td>
<td>76</td>
</tr>
<tr>
<td>5.17</td>
<td>Overview of the combined predictor</td>
<td>78</td>
</tr>
<tr>
<td>5.18</td>
<td>Rate Distortion curve for the mode selection predictor for Scenecut.</td>
<td>80</td>
</tr>
<tr>
<td>6.1</td>
<td>Rate Distortion curve for the proposed algorithm for Coastguard at complexity 50%.</td>
<td>88</td>
</tr>
<tr>
<td>6.2</td>
<td>Rate Distortion curve for the proposed algorithm for Coastguard at complexity 25%.</td>
<td>88</td>
</tr>
<tr>
<td>B.1</td>
<td>PSNR drop</td>
<td>105</td>
</tr>
<tr>
<td>B.2</td>
<td>Rate increase</td>
<td>106</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Huffman coding. Probability of occurrence for the symbols . 15
2.2 Huffman coding. Code words . . . . . . . . . . . . . . 16
2.3 Arithmetic coding . . . . . . . . . . . . . . . . . . . 18
2.4 Arithmetic coding . . . . . . . . . . . . . . . . . . . 18

4.1 Reference software settings . . . . . . . . . . . . . . 30
4.2 Test sequences . . . . . . . . . . . . . . . . . . . . . . . 30
4.3 Rate distortion cost for mode 16x16 . . . . . . . . 42

5.1 Results for the intra predictor . . . . . . . . . . . . . 50
5.2 Results for the intra predictor . . . . . . . . . . . . . 51
5.3 Results for the skip predictor . . . . . . . . . . . . . 55
5.4 Results for the skip or mode 16x16 predictor . . . . 59
5.5 Results for the mode 8x8 predictor . . . . . . . . . . 62
5.6 Results for the submode predictor . . . . . . . . . . . 65
5.7 Results for the RD cost predictor . . . . . . . . . . . 68
5.8 Reference frames . . . . . . . . . . . . . . . . . . . . . . 72
5.9 Summary of which modes are evaluated by the predictors . . 79
5.10 Results for the mode selection predictor . . . . . . . 81
5.11 Results for the mode selection predictor . . . . . . . 82

6.1 Results for the proposed algorithm . . . . . . . . . . 85
6.2 Results for the proposed algorithm . . . . . . . . . . 86
6.3 Results for the proposed algorithm . . . . . . . . . . 87
Chapter 1

Introduction

1.1 Purpose

The purpose with this masters thesis is to optimize mode selection in H.264 in order to reduce the number of modes needed to be evaluated. The work should result in a report and modification of the reference software.

1.2 Project Review

Initially the problem and limitations were discussed with our supervisor at Popwire Technology. A pre-study was then performed, containing information gathering and obtaining knowledge about video coding in general and H.264 and mode selection specifically. Information about video coding and H.264 were obtained mainly through books, while more specific information surrounding mode selection was found in articles. The mode selection algorithms presented in articles were noticeable often only applicable to certain test sequences, because thresholds and other parameters were decided offline to yield the best performance for specific conditions. Therefore a robust and general algorithm with adaptive thresholds soon became our goal.

An extensive statistical analysis was performed in order to obtain information which the algorithm for mode selection could be based on. The statistical analysis and the development of the algorithm for mode selection has been performed using the H.264 reference software\cite{22}. The reference software is extremely slow which have been decreasing work progress. For example encoding a QCIF frame ($176 \times 144$ pixels) takes approximately 10
Another setback with the reference software is that it is not designed for using an effective mode selection since the time for coding is almost the same even if several modes have been discarded. Therefore it is difficult to know how the proposed algorithm actually affects the time of coding because it depends on codec design. However, the average number of evaluated modes gives a good indication of the computational savings.

1.3 Report Outline

Chapter 2 gives an introduction to video coding while in chapter 3 the H.264 standard is discussed more specific. Chapter 4 consists of the statistical analysis that is used for the development of a mode selection algorithm, explained in chapter 5 and 6. Future work is covered in chapter 7. A table of abbreviations can be found at the end of this report.

---

\(^1\) On an Apple G4, 500 MHz, 512 MB SDRAM
Chapter 2

Video Coding in General

2.1 Introduction

An image can be seen as a two dimensional projection from the three dimensional world. The image needs to be sampled in order to be represented digitally. The samples are called pixels and each pixel is represented with an integer number of bits, e.g. 24 bits. The number of pixels is called the resolution of the image. The resolution usually span from $176 \times 144$ for QCIF to $1920 \times 1080$ for HD TV. In video coding these images are referred to as frames and a video sequence consists of a number of frames. The frame rate, i.e. how often there is a new frame, is measured in frames per second (fps) and 25 - 30 fps are common for TV - Broadcast and 7 - 12 fps are common for 3G telephones. Notice that the bitrate required for uncompressed PAL video ($720 \times 576$ pixels) at 25 fps with 24 bits per pixel is nearly 250 Mbit/s. At this bitrate, approximately two and a half minute could be stored on a DVD. In order to be able to store or transmit digital video there is an obvious need for compression. Compression is obtained by removing redundant information. There are three kinds of redundancy: spatial, temporal and statistical.

The temporal redundancy is due to the fact that two consecutive frames often are similar. This fact makes it more effective to code the difference between two frames, referred to as the residual, than coding the frames separately. By performing a motion estimation, i.e. referring to similar areas in previously coded frames, the energy in the residual decreases and the compression performance can increase.
Spatial redundancy arises since images often contain areas with the same or similar pixel values. In other words, nearby pixel values are often highly correlated. The solution is to apply a transform on the residual that decorrelates the data. One common transform is the Discrete Cosine Transform which concentrates the energy in the residual into a few transform coefficients. These coefficients are then quantized in order to represent each sample with a finite number of bits. Further compression is obtained by removing statistical redundancy by performing entropy coding.

More information about video coding in general can be obtained from [1], [2], [3] and [23].

2.2 Color Spaces

2.2.1 RGB

The RGB color space uses the colors Red, Green and Blue to represent each sample in an image. The RGB color space is common for capturing and displaying images. The fact that the three different RGB components usually are regarded as equally important makes it more difficult to obtain compression.

2.2.2 YCbCr

Since the Human Visual System (HVS) is more sensitive to luminance (brightness) than to chrominance (color) a better way to represent an image is to store the luminance component, $Y$, with higher resolution than the chrominance components, $Cb$, $Cr$ and $Cg$. The luminance component is calculated by a weighted average of R, G and B.

$$Y = k_r R + k_g G + k_b B \quad (2.1)$$

where the $k_r$, $k_g$ and $k_b$ are weighting factors.

The color components are calculated as the difference between the R, G and B components and the luminance component, $Y$:

$$Cr = R - Y \quad (2.2)$$

$$Cb = B - Y \quad (2.3)$$
2.3. INTERLACED VIDEO

\[ C_g = G - Y \] (2.4)

The \( C_g \) component is actually redundant and is not necessary to store or transmit. Weighting factors \( k_r = 0.299 \), \( k_g = 0.587 \) and \( k_b = 0.114 \) are often used, which yield the following equations:

\[ Y = 0.299R + 0.587G + 0.114B \] (2.5)
\[ Cr = 0.713(R - Y) \] (2.6)
\[ Cb = 0.564(B - Y) \] (2.7)

**Sampling Formats** As mentioned in section 2.2 each sample is a combination of one luma sample, \( Y \), and two chroma samples, \( Cr \) and \( Cb \). Figure 2.1 shows the sampling formats. The format to the left is referred to as 4:4:4, where the components have the same resolution. Since the eye is more sensitive to luminance than to chrominance formats like 4:2:2 and 4:2:0 are often used. In 4:2:0 the chroma components, \( Cr \) and \( Cb \), have half the resolution compared to the luma component, \( Y \). This sampling format is the most common and is used in DVD and digital television.

![Sampling formats](image_url)

Figure 2.1: **Sampling formats.** White dots represent luminance, while grey and black dots represent chrominance.

2.3 Interlaced video

An interlaced video frame is divided into two fields, sampled at different moments both temporally and spatially. The field consists of samples from
either odd-numbered or even-numbered rows of pixels. Since each field contains half the data the field rate can be twice the frame rate, which gives smoother motion appearances.

2.4 Quality

It is necessary to have a good measure of quality in order to obtain a fair comparison between sequences. There are both objective and subjective quality measures which are described below.

2.4.1 Objective Quality Measures

**PSNR**  Peak Signal to Noise Ratio (PSNR) is widely used as a quality measure. It can be expressed using the logarithm according to equation 2.8.

\[
PSNR_{dB} = 10 \log_{10} \left( \frac{(2^n - 1)^2}{MSE} \right)
\]

where \(n\) is the number of bits used for each sample and MSE (Mean Square Error) is calculated by averaging the sum of squared differences between the current and the reconstructed frame.

The nature of the logarithm implies that a quality degradation of a frame by 50% will result in a PSNR drop of 3 dB. The human eye can notice a difference in PSNR of 0.5 dB. Thus, to maintain the same perceptual quality while processing a frame, the drop in PSNR should not exceed that specific value.

2.4.2 Subjective Quality Measures

The objective quality measure does unfortunately not always correspond to the perceptual quality. Two images with equal objective quality, e.g. same PSNR, can be considered completely different by a subjective observer. An experienced observer and an unexperienced observer often grade the same sequence differently because the experienced observer usually finds known types of artifacts. The rating of an entire sequence is also heavily based on the last moments of viewing, called ”recency effect”. It has been showed that the observer’s viewing environment and state of mind also affect the rating. Subjective quality measures are discussed further in [1] and [2].
2.5 Codec Overview

A Codec consists of an encoder and a decoder and the name is an abbreviation of COder and DECoder. Figure 2.2 gives an overview of a Codec. The encoder compresses a source signal into a bitstream which is stored or transmitted, whilst the decoder reconstructs the signal by decompressing the bitstream. If the original signal and the reconstructed signal are identical the coding process is lossless, otherwise if the reconstructed signal differs from the original the coding process is lossy. In order to achieve necessary compression codecs usually introduce distortion, i.e. lossy coding.

Figure 2.2: Overview of a codec. The encoder represents the original signal with a bitstream for storage or transmission. The bitstream is decoded by the decoder in order to reconstruct the signal.

2.5.1 Encoder

The encoder compresses a source signal for more efficient transmission or storing. Figure 2.3 gives an overview of an encoder. Previous frames are used to perform a motion estimation, which yields motion vectors. These motion vectors are used to make a motion compensated frame. This frame is then subtracted from the current frame and the residual is transformed and quantized. The quantized transform coefficients are entropy coded and transmitted or stored along with the motion vectors found in the motion estimation process. The quantized transform coefficients are also dequantized and inverse transformed in order to obtain reconstructed reference frames.

2.5.2 Decoder

The decoder receives a well-defined bitstream, consisting of entropy coded quantized transform coefficients, motion vectors and header information.
CHAPTER 2. VIDEO CODING IN GENERAL

Figure 2.3: **Overview of an encoder.** Motion compensation constructs a residual containing low energy, which is transformed, quantized and entropy coded into a bitstream. The bitstream is stored or transmitted.

Figure 2.4 gives an overview of a decoder. The bitstream is entropy decoded, dequantized and inverse transformed. The received motion vectors are used to make a motion compensated frame which is added to the inverse transformed frame, which yields the decoded frame.

### 2.6 Predictive Coding

Predictive coding means that instead of coding a signal, $s$, directly, the encoder tries to predict the signal using information of earlier coded signals. This gives $\hat{s}$, an estimate of $s$. Now the difference of $s$ and $\hat{s}$ (called the residual) is calculated and stored. $r = s - \hat{s}$. If the prediction is good, then the residual will be small and only a small amount of data needs to be stored or transmitted. In video coding, a large amount of computation is spent
2.7 MOTION ESTIMATION AND COMPENSATION

Figure 2.4: **Overview of an decoder.** The decoder receives a bitstream, which is entropy decoded, inverse transformed and dequantized in order to reconstruct a residual frame. Motion compensation forms regular frames.

on finding a good prediction. How that works will be described in the next section.

2.7 Motion Estimation and Compensation

In consecutive frames in a movie sequence there is usually considerable temporal redundancy, that is two contiguous frames have a lot in common. Clearly the encoder should take advantage of that. As a first attempt to encode frame nr $i$, use the previous frame as prediction and calculate $r = \text{frame}_i - \text{frame}_{i-1}$. Figure 2.5 shows the residual between frame 5 and 6 of Foreman\(^1\). The residual frame clearly contains less energy than the original frame. The energy in the residual frame arises from e.g. noise, object movement, camera panning or zooming and light changes (shadows etc). If the encoder could estimate how objects moved from one frame to another and compensate for that, the energy in the residual would decrease even more. Motion estimation is the process of finding how pixel values in different areas have moved from one frame to another. The most common way to perform

\(^1\)Common test sequence.
this is by using Block Based Motion Estimation. Here, each frame is divided into blocks of 16x16 pixels, called macroblocks (MB) and motion estimation is performed for each of those. The algorithm can be expressed as:

1. For each macroblock in a frame, find the best 'match' in the previous frame, according to some criteria. A match that minimizes the residual energy in the current MB is a common criteria. Equations 2.9, 2.10 and 2.11 lists a number of energy measures. The offset in x - and y - direction from the 'match' to the MB is called motion vector (mv).

2. Subtract the best candidate from the original block to form the residual. This is called motion compensation. Encode the residual and store it together with the motion vector.

![Figure 2.5: Residual between frame five and six for Foreman.](image)

The decoder adds the residual to the macroblock in the previous frame pointed out by the motion vector. This gives a reconstructed version of the original macroblock. The same decoding procedure is performed in the
encoder as well. This is to make sure that the encoder and the decoder uses identical reference frames for future motion compensation.

Figure 2.6 shows the motion compensated residual. As expected, the energy is considerably lower than without motion compensation.

Figure 2.6: Motion compensated residual between frame five and six for Foreman.
Different types of energy measurements  The type of energy measure function will affect computational complexity and accuracy of the motion estimation process. The measurements MSE (Mean Squared Error), MAE (Mean Absolute Error) and SAE (Sum of Absolute Errors) are presented below. SAE, also known as SAD (Sum of Absolute Differences), is the most common used due to its computational simplicity.

\[
MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C(i, j) - R(i, j))^2  \tag{2.9}
\]

\[
MAE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i, j) - R(i, j)|  \tag{2.10}
\]

\[
SAE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i, j) - R(i, j)|  \tag{2.11}
\]
2.8 Transform and Quantization

2.8.1 Transform Coding

Transform coding is widely used in video coding and the most common transform is the two dimensional discrete cosine transform (2D-DCT). For image coding the discrete wavelet transform (DWT) is also used. The transformation does not achieve any compression by itself though. The transform represents data in another way, which makes it possible to remove spatial correlation and the energy will be concentrated in a few significant coefficients. Insignificant coefficients can then be discarded without affecting the image quality. The transform is reversible i.e. there is an inverse transform that transforms back to the spatial domain.

**Discrete Cosine Transform** The 2D-DCT can be obtained by performing a 1D-DCT on the rows followed by a 1D-DCT on the columns and is usually implemented using matrix multiplication. The fact that the DCT transforms data effectively and that it rather easily can be implemented in both software and hardware makes it the most used transform for image and video coding. The DCT is a block transform, i.e. the transform is applied on blocks of pixels instead of the entire image. Because the computational complexity increases between squared and cubic with block size, usually small blocks like 8×8 pixels are used. If the image samples is represented with \( f_{i,j} \) the DCT coefficients can be calculated according to equation 2.12.

\[
F_{x,y} = \sum_{i=0}^{N} \sum_{j=0}^{N} \frac{C(x)C(y)}{4} f_{i,j} \cos \left( \frac{(2i + 1)x\pi}{16} \right) \cos \left( \frac{(2j + 1)y\pi}{16} \right) \tag{2.12}
\]

The inverse DCT is given by equation 2.13

\[
f_{i,j} = \frac{C(x)C(y)}{4} \sum_{i=0}^{N} \sum_{j=0}^{N} F_{x,y} \cos \left( \frac{(2i + 1)x\pi}{16} \right) \cos \left( \frac{(2j + 1)y\pi}{16} \right) \tag{2.13}
\]

where the \( C(x) \) and \( C(y) \) are constants:

\[
C(n) = \begin{cases} 
\frac{1}{\sqrt{2}}, & n = 0 \\
1, & n \neq 0
\end{cases}
\]
2.8.2 Quantization

Scalar Quantization  The transform coefficients may assume any value which makes entropy coding difficult. The transform coefficients are therefore quantized, i.e. rounded to certain levels, see figure 2.7. These levels are parted with the size of the quantization step $\Delta q$. As mentioned in chapter 2.8.1 transform coefficients that are insignificant are quantized to zero and are unnecessary to transmit or store.

Figure 2.7: Linear quantizer. Input values are mapped on discrete levels.

Vector Quantization  In vector quantization a block of samples is mapped on to a single code word. The block is compared with blocks in a predetermined code book and the index representing the best match is transmitted or stored. The decoder, which has the same code book, receives an index and returns a block of samples. The quantization is of course lossy because there will not usually be perfect matches in the code book. In order to make good matches and minimize the distortion the code book usually needs to be quite large. There are some difficulties with a large code book like how to store it and how to perform the complex search for the best match.
2.9 Entropy Coding

In order to store or transmit e.g. quantized transform coefficients efficiently, further compression must be done. By considering the statistics of a source it is possible to obtain compression by removing statistical redundancy. Entropy coding is lossless, i.e. the decompressed data is identical to the original data. More information about entropy coding can be obtained from [1], [2] and [23].

2.9.1 Huffman Coding

The huffman code is a Variable Length Code (VLC) which means that symbols may be mapped on to code words with different number of bits. The idea with Huffman encoding is that symbols that occur more frequently are coded with shorter code words. This means that the probability of the occurrence of each symbol must be known.

The huffman code is constructed by building a tree, called huffman tree where each symbol corresponds to a leaf in the huffman tree. The two symbols with lowest probability is combined into a new node in the tree. The probability of this node is the sum of the probability for two merged symbols. The two branches from the new node is assigned with 0 and 1 respectively. This procedure, combining the two leaves and/or nodes with lowest probability, is then repeated until the root node is reached. The probability of the root node is 1 because it is the sum of the probabilities for all symbols. The code word for each symbol is obtained by starting at the root and appending the value assigned to each branch until the leaf node is reached.

Example 1 (Huffman) A source consists of an alphabet with the symbols \{A, B, C, D\}, with probability of occurrence according to table 2.1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.6</td>
</tr>
<tr>
<td>B</td>
<td>0.2</td>
</tr>
<tr>
<td>C</td>
<td>0.1</td>
</tr>
<tr>
<td>D</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2.1: Huffman coding. Probability of occurrence for the symbols
A huffman tree, see figure 2.8, is formed according to the procedure explained above and the code word for each symbol is easily obtained from the figure. The code words are presented in table 2.2.

![Huffman Tree Diagram]

Figure 2.8: Huffman tree

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Code word</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>001</td>
</tr>
<tr>
<td>D</td>
<td>000</td>
</tr>
</tbody>
</table>

Table 2.2: Huffman coding. Code words

The average length of the code words is obtained by averaging the code word lengths weighted with their probability. In this example the average code word length is 1.6 bits / symbol which should be compared to 2 bits / symbol without entropy coding.

### 2.9.2 Arithmetic Coding

In arithmetic coding a sequence of symbols is mapped on to a code word. This approach often gives better compression performance than variable length coding.
2.9. ENTROPY CODING

The idea with arithmetic coding is to represent the sequence with an interval, which will determine the code word for the sequence. To find this interval the probability of occurrences for each symbol must be known. A probability interval $[0, 1]$ is divided into subintervals according to the probability of occurrence of the symbols. The subinterval associated with the first symbol in the sequence is then regarded as the new interval. This interval is divided into subintervals with the same proportions as the original interval was divided into. See figure 2.9. The subinterval in the new interval associated with the next symbol in the sequence is then regarded as the new interval. This procedure is repeated until a certain number of symbols are processed.

The sequence can be represented by any fractional number in the final interval. It can be showed that the length, $l$, of the code word is at most

$$l = \lceil \log(p) \rceil + 1$$

(2.14)

where $p$ is the length of the final interval that represents the sequence.

The codeword is obtained by truncating the binary representation of any number in the interval to $\lceil \log(p) \rceil + 1$ bits.

Example 2 (Arithmetic) The alphabet consists of the symbols \{A, B, C, D\}, with probability of occurrence according to table 2.3. The sequence that is being encoded is \{B, B, A, C\}. An interval, see table 2.4, for the sequence is decided according to the procedure explained above and according to equation 2.14. The final interval is $[0.7392, 0.7416]$ which will need 10 bits to code. The ten first bits in the binary representation of a value in the final interval is the codeword. For example: $0.7404_{10} = .101110110..._2$ which yields $101110110$ as the codeword. Ten bits for coding the sequence corresponds to 2.5 bits / symbol in average. The average will decrease when coding larger sequences.
Table 2.3: **Arithmetic coding.** Probability of occurrence for symbols A, B, C and D

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.6</td>
</tr>
<tr>
<td>B</td>
<td>0.2</td>
</tr>
<tr>
<td>C</td>
<td>0.1</td>
</tr>
<tr>
<td>D</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Figure 2.9: **Arithmetic coding - Probability intervals for the symbols**
If B is the first symbol in the sequence then the interval [0.6 0.8] is regarded as the new interval. This interval is divided into subintervals according to the probability of occurrences of the symbols.

Table 2.4: **Arithmetic coding.** Probability intervals for the coded sequence.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>[0.6 0.8]</td>
</tr>
<tr>
<td>BB</td>
<td>[0.72 0.76]</td>
</tr>
<tr>
<td>BBA</td>
<td>[0.72 0.744]</td>
</tr>
<tr>
<td>BBAC</td>
<td>[0.7392 0.7416]</td>
</tr>
</tbody>
</table>
Chapter 3

H.264

This chapter treats the features of H.264. A number of improvements make H.264 twice as efficient compared to other codecs. More information about H.264 can be obtained from [1] and [8].

3.1 Structure

A frame consists of a number of slices each containing a number of macroblocks. There are five types of slices: I (Intra), P (Predicted), B (Bi-predictive), SP (Switching P) and SI (Switching I) and a frame can contain a mixture of these types. The main idea with slices is that parts of a frame can be coded independent of each other. This makes it possible to code two slices simultaneously on dedicated hardware.

Figure 3.1: QCIF Frame consisting of two slices.
Each slice consists of a number of macroblocks that are 16x16 pixels. H.264 uses two different block sizes for intra coding namely 16x16 and 4x4. For inter coding (P or B frames) the macroblock can be divided into partitions: 16x8, 8x16 and 8x8 pixels. Each 8x8 block can be further divided into subpartitions: 8x4, 4x8 and 4x4. These partitions are illustrated in figure 3.3.

Figure 3.2: **Macroblock and subblock partitions** Each subblock in macroblock partition 8x8 can be divided into subblock partitions 8x8, 8x4, 4x8 and 4x4.
Figure 3.3: **Macroblock and subblock partitions** An example of macroblock partitioning for a frame in the test sequence Carphone. The crossed macroblocks represent skipped macroblocks. Notice that subpartitioning are chosen where there are a lot of details while larger blocks and skip are chosen where there are limited amount of details.
3.2 Profiles

There are four profiles in H.264: Baseline, Main, Extended and High [1].

- **Baseline**: The baseline profile handles intra coding and P-slice inter coding. Entropy coding is performed through CAVLC, Context-based Adaptive Variable Length Coding. The primary application of the baseline profile is low delay wireless communication.

- **Main**: In the main profile inter coding using B-slices is supported. Interlaced video can be handled and entropy coding can be performed using CABAC, Context-based Adaptive Binary Arithmetic Coding. The main profile is suitable for television broadcasting and video storage.

- **Extended**: Interlaced video and CABAC are not supported in the extended profile, however it has improved error resilience and uses efficient switching between coded bitstreams (SP- and SI-slices). These features makes this profile useful for streaming.

- **High**: This profile supports sampling formats 4:2:2 and 4:4:4.

3.3 Intra Coding

An intra frame is coded independently of other frames. A prediction for each macroblock is therefore calculated using nearby already coded macroblocks. A low energy residual is formed by subtracting the prediction from the original macroblock. There are four prediction modes for intra 16x16: horizontal, vertical, DC (mean) and plane. The prediction modes for Intra 16x16 are illustrated in figure 3.4. For intra 4x4 there are nine prediction modes: horizontal, vertical, DC, diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left and horizontal-up. The prediction modes for intra 4x4 are illustrated in figure 3.5.
Figure 3.4: **Prediction modes for intra 16x16 prediction** V and H represent pixels to the left and above the macroblock respectively.
Figure 3.5: Prediction modes for intra 4x4 prediction
3.4 Motion Estimation and Compensation

3.4.1 Multiple reference frames

In H.264 there are several reference frames stored in a buffer for use in the motion compensation procedure. This requires more memory in the encoder/decoder but it also increases compression performance, since it is not certain that the best match will be obtained in the most previous frame. The buffer can also store older frames, e.g. the last frame before a scene change. For example, let a movie sequence consist of clips from two different scenes, scene1 and scene2 and that they are arranged as scene1 - scene2 - scene1. That is the movie starts with scene1, switches to scene2 and ends with scene1. Assume that the first frame of the second scene1 is to be encoded. If there was only one reference frame available for motion estimation, there would be a poor match since the reference frame belongs to scene2. But if the last frame of the first occurrence of scene1 is stored in the reference frame buffer, the motion estimation would probably find a match in that frame, just as if there were no scene change.

3.4.2 Block Partitioning

One major advantage in H.264 is that motion estimation is not performed only on blocks of 16x16 pixels. One macroblock can be divided into partitions of 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 pixels. Figure 3.3 shows the different partitions. Smaller blocksizes will naturally find better matches in areas of high complexity/motion and therefore produce a residual with lower energy. On the other hand, smaller blocks means that more motion vectors need to be used. There must therefore be a tradeoff between distortion and compression performance. This tradeoff, called Rate-Distortion-Optimization, can be expressed as an optimization problem:

Choose the block partitioning that fulfills the following equation:

\[ \min D \quad \text{subject to} \quad R < R_t \]  \hspace{1cm} (3.1)

Where \( D \) is the distortion, \( R_t \) the target bitrate and \( R \) the number of bits needed to represent the coded residual and the motion vectors. Equation 3.1 can be solved introducing a lagrangian parameter. Then the optimization
problem becomes to minimize the rate distortion cost function, $J$, according to equation 3.2.

$$J = D + \lambda R$$  \hspace{1cm} (3.2)

Where the lagrangian parameter $\lambda$ has been empirically determined to be $0.85 \times 2^{(QP-12)/3}$, where QP is the quantization parameter. This is further explained in [26].

Every possible partitioning of a macroblock can be evaluated, and the combination with the lowest Rate-Distortion cost is selected and coded.

### 3.4.3 Subpixel Motionsearch

It is not likely that a moving object moves exactly an integer number of pixels from one frame to another. To find a better match, H.264 supports subpixel motion estimation down to a quarter pixel accuracy. How to perform this is up to the designer of the codec. One possible solution is to create the subpixel image by interpolation between integer pixels and perform the motion estimation on this new, higher resolution image. Another possibility is to start with a regular integer motion estimation and when the best integer match is found, the motion estimation process continues with half pixel accuracy. This is called halfpixel motion estimation and is done by interpolation with the nearby integer pixels. The next and final step in the process is quarterpixel motion estimation, with quarter pixel accuracy. The best match together with the corresponding motion vector is used for motion compensation. Figure 3.6 shows the relationship between fullpixel, halfpixel and quarterpixel.

### 3.5 Deblocking Filter

When using a block based transform the appearance of sharp block edges is a common artifact. In order to avoid this a filter is used to give smooth edges. The deblocking filter is applied on the reconstructed image in both the encoder and decoder. By removing the blocking effects in the reconstructed frame in the encoder a better motion estimation can be achieved than when using the unfiltered reconstructed frame.
3.6 Transform and Quantization

3.6.1 Transform Coding

Like other common codecs (MPEG-1, MPEG-2, H.261 and H.263) H.264 uses DCT for transformation. But in difference from the others H.264 uses an integer approximation to the DCT described in section 2.8.1. Using integer arithmetics the DCT/IDCT can be performed using only additions, shifts and scalings. To increase performance the scaling is incorporated in the quantization process. This is thus a computationally efficient transform without significant loss of accuracy compared to the original DCT.

3.6.2 Quantization

Quantization of a coefficient $Y_{i,j}$ can normally be expressed on the form $Z_{i,j} = \text{round}(Y_{i,j}/QP)$. Division is normally a computationally expensive operation. To avoid division, quantization in H.264 utilizes multiplications and
arithmetic shifts. This can be expressed as \( Z_{i,j} = \text{round}(W_{i,j} \times MF/2^{qbits}) \), where \( W_{i,j} \) is the unscaled DCT coefficient and \( qbits \) is dependent of the current QP. \( MF \) is a multiplication factor containing among other things the scaling factor from the DCT.

### 3.7 Entropy Coding

There are two different entropy coding methods in H.264; Context-based Adaptive Variable Length Coding (CAVLC) and Context-based Adaptive Binary Arithmetic Coding (CABAC). CAVLC and CABAC are not further utilized in this thesis but can be read about in [8].
Chapter 4

Statistical Analysis

4.1 Introduction

This chapter contains a statistical analysis of mode selection and some macroblock measures. In order to achieve a fair analysis many different test sequences and quantization parameters have been used.

The test sequences used are standard sequences like: Carphone, Coastguard, Claire and Foreman. Some sequences, like Claire, or parts of sequences contain very limited movement. Other sequences like Foreman and Coastguard consist of a great amount of movement. Since scene cuts are commonly used in videos, especially in music videos, a test sequence with scene cuts has also been used, referred to as Scenecut.

A range of different QP values have been used. The smallest QP used is 16 which corresponds to the highest quality, but also the highest bitrate. The quantization parameter is increased with steps of four until maximum is reached at QP = 40.

Version 8.6 of the reference software[22] has been used. The settings are presented in table 4.1. Data, like motion vectors and macroblock partitions, are extracted from the codec and further analyzed in Matlab. The work has been performed on an Apple G4, 500 MHz with 512 MB SDRAM.

4.2 Mode Statistics

This section contains an analysis on the selection of modes. The aim with the analysis is to obtain information about which mode that is chosen under
different conditions. As explained earlier a partitioning is selected for each macroblock. These partitionings are now referred to as modes. There are two intra modes, 16x16 and 4x4. There are four inter modes 16x16, 16x8, 8x16 and 8x8. Mode 8x8 can be divided further into submodes 8x8, 8x4, 4x8 and 4x4. There is also a skip mode where pixels are copied directly from the previous frame. Each mode is evaluated and the mode that minimizes the rate-distortion cost function is regarded as the best mode.

### 4.2.1 Selection of Best Mode

The first stage is to examine the probability of occurrence for each mode. This should give a hint of the importance of each mode under different conditions. Hopefully some modes can be considered as less important, or even discarded. The probability of occurrence for each mode for some sequences

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Baseline</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>30</td>
</tr>
<tr>
<td>Resolution</td>
<td>176x144 (QCIF)</td>
</tr>
<tr>
<td>Number of reference frames</td>
<td>10</td>
</tr>
<tr>
<td>Search range</td>
<td>16</td>
</tr>
<tr>
<td>Loop filter</td>
<td>enable</td>
</tr>
<tr>
<td>Skip frame</td>
<td>disable</td>
</tr>
<tr>
<td>Rate control</td>
<td>disable</td>
</tr>
<tr>
<td>Intra period</td>
<td>Only first frame</td>
</tr>
</tbody>
</table>

**Table 4.1: Reference software settings**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>Fast object motion with part of background</td>
</tr>
<tr>
<td>Claire</td>
<td>&quot;Talking head&quot;</td>
</tr>
<tr>
<td>Coastguard</td>
<td>Object translation and panning</td>
</tr>
<tr>
<td>Foreman</td>
<td>Object translation and panning</td>
</tr>
<tr>
<td>Scenecut</td>
<td>Irregular movement and scenecuts</td>
</tr>
</tbody>
</table>

**Table 4.2: Test sequences**
is shown in figures 4.1 and 4.2.

The term *probability* used above is not exactly correct since there is nothing stochastic about the occurrence of the different modes. The correct term is rather *relative frequency*. But for simplicity, especially when talking about conditional probability, the term probability will be used during the rest of this thesis even though the slight misuse.

![Diagram showing mode statistics](image)

Figure 4.1: **Probability of optimal modes** for Claire

**Results** A comparison of figures 4.1 and 4.2 shows that the probability of skip for Claire is much higher than for Foreman and that the probability increases with increasing QP.

For mode 8x8, which includes submodes 8x8, 8x4, 4x8 and 4x4, the relationships are reverse. That is the probability of mode 8x8 decreases with increasing QP and the probability of mode 8x8 is higher for Foreman than for Claire.
CHAPTER 4. STATISTICAL ANALYSIS

Figure 4.2: Probability of optimal modes for Foreman

Observation 1 The probability of occurrence of the different modes strongly depend on the quantization parameter, QP. A low QP increases the probability for modes with small block sizes and vice versa.

Observation 2 The probability of occurrence of the different modes strongly depend on the test sequence. Sequences like Claire contains more modes with large block sizes and skip than for example Foreman.

4.2.2 Influence of Mode in Earlier Macroblocks

Since there are temporal correlation (between frames) the best mode chosen for a macroblock ought to be correlated with the best mode chosen for the macroblock in the same position in the previous frame. Figure 4.3 shows that the probability of classifying a certain mode as optimal increases if the macroblock in the same position in the previous frame were classified as that particular mode.
Observation 3 *The probability of occurrence of a mode depends on the mode chosen for the macroblock in the same position in previous frame(s).*

Figure 4.3: **Temporal correlation for Foreman, QP 24.** The diamonds corresponds to the probability of choosing each mode as optimal. Considering temporal correlation the conditional probability will be higher, which is illustrated in this figure by the squares.

There is also spatial correlation (in a frame) which should imply that the mode of macroblocks that are close to the current macroblock are correlated. Figure 4.4 shows that the probability of classifying a certain mode as optimal increases if the macroblock above and to the left of the current macroblock was classified as that particular mode.

Observation 4 *The probability of occurrence of a mode depends on the mode chosen for the macroblocks close to the current macroblock.*
Figure 4.4: Spatial correlation for Carphone, QP 32. The diamonds correspond to the probability of choosing each mode as optimal. Considering spatial correlation the conditional probability will be higher, which is illustrated in this figure by the squares.

4.2.3 Reference Frames

The H.264 standard allows motion estimation in more than one reference frame to obtain the optimal mode. Figure 4.5 shows in which reference frame the best mode is found and it clearly shows that the best match is often found in frames that are close to the current frame. Therefore it can be unnecessary to use a larger number of reference frames.

Observation 5 The optimal mode is often found in frames that are close to the current frame.

Although most optimal modes are found in frames close to the current frame limiting the number of reference frame can decrease the quality. An analysis of the change in quality when alternating the number of reference
4.2. MODE STATISTICS

frame shows that the quality decreases more rapidly for each removed reference frame.

**Observation 6** The quality decreases more rapidly for each removed reference frame.

![Figure 4.5: Reference frames. Probability of finding the optimal mode in various reference frame](image-url)
4.3 Macroblock Measures

This section discusses some measures on macroblocks that could help explain the characteristics of a sequence. Hopefully some measures will give information about which modes that are more probable than others to be the optimal.

4.3.1 Energy Estimating Measurers

The following measures give an approximation of the energy in the residual, which can be used to classify which mode is most likely to be chosen. The measures can be performed on macroblocks and/or subblocks and are obtained without the use of motion estimation, transformation etc. since they are only based on the pixel values in the current and, if needed, the most recent reference frame.

SAD  As mentioned earlier, the most common measure used to estimate the energy in the residual is SAD, Sum of Absolute Differences. A slight disadvantage with SAD is that it is too sensitive to constant fading of the luminance. A constant fading, i.e. all values are changed with the same amount, is actually just a change of the DC level and does not affect the difficulty to code the residual.

Figure 4.6 shows the distribution of SAD for every mode selected as the best. The X in each figure marks the mean value of SAD. The SAD values for modes skip, mode 16x16 and mode 8x8 seem to be distributed around the same mean value, approximately 1100. It seems there is no obvious way to distinguish the different modes using only SAD.

Pixdiff  In order to manage the fading problem with SAD another measure has been developed and evaluated. This measure is referred to as pixdiff and represents an approximation of the derivative of SAD. Pixdiff requires however slightly more computations than SAD.

The residual, \( Res \), is formed by subtracting the reconstructed frame from the current frame. \( Diff_x \) and \( Diff_y \) is the sum of the difference of adjacent pixel values in the x- and y-direction respectively.

\[
Diff_x = \sum_{i=0}^{N-2} \sum_{j=0}^{N-1} |Res(i + 1, j) - Res(i, j)|
\]
4.3. MACROBLOCK MEASURES

\[
\text{Diff}_y = \sum_{i=0}^{N-1} \sum_{j=0}^{N-2} |\text{Res}(i, j + 1) - \text{Res}(i, j)|
\]  

(4.2)

Pixdiff is then obtained by averaging \( \text{Diff}_x \) and \( \text{Diff}_y \) and adding the first value in the residual, which in some way represents a DC-level.

\[
\text{Pixdiff} = \frac{\text{Diff}_x + \text{Diff}_y + \text{Res}(0,0)}{2}
\]

(4.3)

Figure 4.6 shows the distribution of pixdiff for every mode selected as the best. The X marks the mean value of pixdiff. In difference from SAD, macroblocks divided into smaller partitions tends to have larger values than macroblocks with larger partitions. Let us take a look at the differences in a more mathematical way. Let \( X \) be a stochastic variable describing the distribution of skipped macroblocks. In the same way let \( Y \) and \( Z \) denote
stochastic variables for mode 16x16 and mode 8x8. For simplicity the distribution functions are approximated by the normal distribution. Now let us try the hypothesis that a large pixdiff implies that it is higher probability for a macroblock to be selected as mode 8x8 rather than of mode 16x16. A 95% confidence interval is used to try the hypothesis. According to statistical theory the confidence interval for the difference between two average values is approximated using:

\[ I_{m_Z-m_Y} = m_Z - m_Y \pm \lambda_{\alpha/2} \sqrt{\frac{s_Y}{n_Y} + \frac{s_Z}{n_Z}} \]  

(4.4)

where 1 - \(\alpha\) is the degree of confidence.

\[ I_{m_Z-m_Y} = m_Z - m_Y \pm 1.96 \sqrt{\frac{s_Y}{n_Y} + \frac{s_Z}{n_Z}} = 3339 - 1984 \pm 1.96 \times 43.04 = [1270, 1439] \]

The same equation yields \(I_{E[Z-X]} = [2071, 2235]\) and \(I_{E[Y-X]} = [765, 832]\). The computed intervals does not include zero, which means that the hypothesis is verified since \(n_X \approx n_Y \approx 10000\) is sufficiently large. Similar results are obtained using other movies and QP:s. This leads to the following observations.

**Observation 7**  A small value of pixdiff indicates that the macroblock is most likely to be skipped. A large value of pixdiff on the other hand indicates that the macroblock most likely is going to be selected as mode 8x8. Finally a macroblock with a medium large pixdiff is probably going to be selected as mode 16x16.

**Observation 8**  Skip and mode 16x16 are difficult to separate using pixdiff.

Figures 4.8(a) and 4.8(b) show the mean values of pixdiff for Foreman and Claire respectively. It is clear that the mean value is quite larger for the sequence containing a fair part of movement, i.e. Foreman, than for a quiet sequence like Claire.

**Observation 9**  Pixdiff is larger for sequences containing movements than for sequences with limited movement.
4.3. MACROBLOCK MEASURES

Figure 4.7: \textbf{Pixdiff} distribution for Foreman 40

Figure 4.8: \textbf{Mean pixdiff} for various QP
Intra  Pixdiff can also be calculated on the original image instead of using the residual. This measurement may be used for prediction of best Intra mode. Figure 4.9 shows the distribution of pixdiff for Intra 16x16 and Intra 4x4.

Observation 10  Pixdiff in a macroblock should be able to be used to predict the best Intra mode. A large value of pixdiff indicates that the macroblock most likely is going to be selected as intra mode 4x4.

\[
\sigma^2 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (Res(i, j) - \overline{Res})^2
\]  (4.5)
where $M \times N$ is the blocksize.

The variance measure is comparable to SAD in order to predict which mode to be selected. Due to its computational complexity the variance is not further explored.

### 4.3.2 Rate Distortion Cost

The RD cost for a macroblock is obtained when a mode is evaluated. This means that RD costs from already tested modes might be used to decide which other modes that need to be fully tested, read more in [4]. The cheapest inter mode to test is mode 16x16, which makes it a good candidate for pre processing. Performing a full motion estimation on mode 16x16 should give a hint whether mode 16x8, mode 8x16 or mode 8x8 (including all sub modes) should be further evaluated. Intuitively, if the RD cost for 16x16 is small, a good match has been found and small modes like 8x8, 8x4, 4x8 and 4x4 is unnecessary to evaluate. If, on the other hand, the RD cost is high, a better match might be found using smaller modes. Denote mode 16x8 and mode 8x16 as large modes and mode 8x8 (including all submodes) as small modes. Then an interval of confidence can be calculated, trying the hypothesis that the mean value of the RD cost for mode 16x16 when large modes are obtained as best, is smaller than the mean value of the RD cost for mode 16x16 when small modes is best. Table 4.3 lists the mean value of RD cost for mode 16x16 when modes 16x8 and 8x16 respectively mode 8x8 is obtained as best mode. The rightmost column contain the interval of confidence (95%). Calculation of the rate distortion cost is discussed in section 3.4.2. None of the intervals contain zero, that is the hypothesis is verified. This leads to the the following observation.

**Observation 11** If the rate distortion cost for mode 16x16 is small, there is a high probability that large modes like mode 16x16, mode 16x8 and mode 8x16 is going to be selected as the best mode. On the other hand, if the rate distortion cost for mode 16x16 is high, mode 8x8 will probably be selected as best mode.
### Table 4.3: Rate distortion cost for mode 16x16.

Mean value of RD cost for 16x16 for large respectively small modes.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Large blocks</th>
<th>Small blocks</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>12</td>
<td>25</td>
<td>[11 16]</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>14</td>
<td>31</td>
<td>[14 20]</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>16</td>
<td>39</td>
<td>[19 27]</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>24</td>
<td>48</td>
<td>[19 29]</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>34</td>
<td>67</td>
<td>[26 41]</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>4</td>
<td>11</td>
<td>[5 9]</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>6</td>
<td>18</td>
<td>[6 18]</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>10</td>
<td>21</td>
<td>[8 15]</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>17</td>
<td>37</td>
<td>[13 27]</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>29</td>
<td>61</td>
<td>[13 50]</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>31</td>
<td>44</td>
<td>[10 16]</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>32</td>
<td>46</td>
<td>[10 18]</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>34</td>
<td>55</td>
<td>[17 25]</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>46</td>
<td>72</td>
<td>[19 32]</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>64</td>
<td>105</td>
<td>[29 53]</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>30</td>
<td>34</td>
<td>[1 7]</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>24</td>
<td>39</td>
<td>[11 17]</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>25</td>
<td>45</td>
<td>[16 23]</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>31</td>
<td>55</td>
<td>[20 29]</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>42</td>
<td>80</td>
<td>[28 47]</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>62</td>
<td>220</td>
<td>[136 181]</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>85</td>
<td>242</td>
<td>[130 183]</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>108</td>
<td>276</td>
<td>[136 200]</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>134</td>
<td>314</td>
<td>[146 215]</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>158</td>
<td>390</td>
<td>[187 275]</td>
</tr>
</tbody>
</table>
4.4 List of Observations

<table>
<thead>
<tr>
<th>Observation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Observation 1</strong></td>
<td>The probability of occurrence of the different modes strongly depend on the quantization parameter, QP. A low QP increases the probability for modes with small block sizes and vice versa.</td>
</tr>
<tr>
<td><strong>Observation 2</strong></td>
<td>The probability of occurrence of the different modes strongly depend on the test sequence. Sequences like <em>Claire</em> contains more modes with large block sizes and skip than for example <em>Foreman.</em></td>
</tr>
<tr>
<td><strong>Observation 3</strong></td>
<td>The probability of occurrence of a mode depends on the mode chosen for the macroblock in the same position in previous frame(s).</td>
</tr>
<tr>
<td><strong>Observation 4</strong></td>
<td>The probability of occurrence of a mode depends on the mode chosen for the macroblocks close to the current macroblock.</td>
</tr>
<tr>
<td><strong>Observation 5</strong></td>
<td>The optimal mode is often found in frames that are close to the current frame.</td>
</tr>
<tr>
<td><strong>Observation 6</strong></td>
<td>The quality decreases more rapidly for each removed reference frame.</td>
</tr>
<tr>
<td><strong>Observation 7</strong></td>
<td>A small value of pixdiff indicates that the macroblock is most likely to be skipped. A large value of pixdiff on the other hand indicates that the macroblock most likely is going to be selected as mode 8x8. Finally a macroblock with a medium large pixdiff is probably going to be selected as mode 16x16.</td>
</tr>
<tr>
<td><strong>Observation 8</strong></td>
<td>Skip and mode 16x16 are difficult to separate using pixdiff.</td>
</tr>
<tr>
<td><strong>Observation 9</strong></td>
<td>Pixdiff is larger for sequences containing movements than for sequences with limited movement.</td>
</tr>
<tr>
<td><strong>Observation 10</strong></td>
<td>Pixdiff in a macroblock should be able to be used to predict the best Intra mode. A large value of pixdiff indicates that the macroblock most likely is going to be selected as intra mode 4x4.</td>
</tr>
<tr>
<td><strong>Observation 11</strong></td>
<td>If the rate distortion cost for mode 16x16 is small, there is a high probability that large modes like mode 16x16, mode 16x8 and mode 8x16 is going to be selected as the best mode. On the other hand, if the rate distortion cost for mode 16x16 is high, mode 8x8 will probably be selected as best mode.</td>
</tr>
</tbody>
</table>
Chapter 5

Optimization of Mode Selection

5.1 Introduction

This chapter provides ideas about how to select and evaluate a subset of modes without performing a complete evaluation of all modes. As mentioned earlier a computational burdensome motion estimation is performed for every inter mode. For intra modes there are no motion estimation performed but because of the numerous kinds of predictions the computational burden becomes high as well.

In order to avoid these computations for every mode several predictors have been implemented based on the results from the statistical analysis, see chapter 4.

The various predictors and results for these are presented below. These are later combined in order to form the proposed algorithm for mode selection. The results are presented as either a PSNR degradation, stated in dB, or a percentage rate increase. The calculation of these measurements is extensively explained in Appendix B. As stated in section 2.4.1 the limit for perceptual image quality degradation is 0.5 dB.

5.2 Intra Mode Predictors

For intra there is a total of thirteen prediction modes. The nine prediction modes for intra 4x4 prediction and the four prediction modes for intra 16x16 modes are by us considered equally computational constraining. The proposed predictor in this section describes ideas how to discard some of these
CHAPTER 5. OPTIMIZATION OF MODE SELECTION

modes. The average number of evaluated prediction modes are used in order to describe the computational complexity for intra, where a complexity of 100% corresponds to evaluate all prediction modes.

The intra predictor consists of two parts: One part that decides if intra 4x4 or intra 16x16 could be skipped and one part that discards RD cost calculations for some prediction modes for intra 4x4.

5.2.1 Intra 16x16 or Intra 4x4 Predictor

As mentioned in section 4.3, pixdiff for the macroblock can be used to predict if intra 16x16 or intra 4x4 is unnecessary to evaluate. This prediction is performed by comparing pixdiff with two thresholds, one for each group of intra modes. The threshold for intra 16x16 is based on the mean values of pixdiff when intra 16x16 is the best mode. The threshold for intra 4x4 is decided in an analogous way. Figure 5.1 shows an overview of the intra 16x16 or intra 4x4 predictor.
Figure 5.1: **Overview of the intra 16x16 or intra 4x4 predictor.** Deciding whether intra 16x16 or intra 4x4 can be discarded.
5.2.2 Intra 4x4 Predictor

As stated earlier there are nine prediction modes for intra 4x4. In order to reduce the computational load for intra 4x4 only a subset of these nine prediction modes are evaluated with full rate distortion calculation. First all predictions are formed and SAD between the predictions and the original block is used as an approximation of the energy. Then a full evaluation is performed on the k predictions with the lowest SAD, where k is set according to the demand of quality versus speed. Figure 5.2 shows an overview of the intra 4x4 predictor.

Figure 5.2: Overview of Intra 4x4 predictor. The energy for all prediction modes are calculated and a full rate distortion optimization is performed on the k modes with lowest energy.
5.2.3 Combined Intra Predictor

The intra predictors described above can with advantage be combined to form a complete intra predictor. Results of this combined predictor is presented below.

5.2.4 Results

Tables 5.2 and 5.1 show the results of the intra predictor. Four different complexity targets have been used: 100%, 75%, 50% and 25%. This corresponds to evaluate 13, 9.75, 6.5 and 3.25 prediction modes in average. Complexity target 100% is used as reference for PSNR and rate comparison.

The combined intra predictor performs very well. According to the tables there is hardly no degradation of PSNR or increase of rate. At complexities below 25 % quality is rapidly decreasing and is therefore not used.
Table 5.1: Results for the intra predictor. Results for various sequences and Qp for the intra predictor. The four right columns represent complexity targets of 100%, 75%, 50% and 25% respectively. 100 % is used as reference for PSNR comparison. A negative $\Delta$PSNR means a degradation in quality. Note that some PSNR values are positive. This is because the margin of error in the PSNR calculations is larger than the actual PSNR drop.
## 5.2. INTRA MODE PREDICTORS

Table 5.2: **Results for the intra predictor.** Results for various sequences and Qp for the intra predictor. The four right columns represent complexity targets of 100%, 75%, 50% and 25% respectively. 100 % is used as reference for rate comparison. A positive $\Delta \text{Rate}$ means that more bits is needed to represent the sequence.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>Rate (kbit/s)</th>
<th>$\Delta \text{Rate}$</th>
<th>$\Delta \text{Rate}$</th>
<th>$\Delta \text{Rate}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>1225</td>
<td>0.015</td>
<td>0.026</td>
<td>0.086</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>883</td>
<td>-0.112</td>
<td>-0.070</td>
<td>0.059</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>626</td>
<td>-0.140</td>
<td>-0.034</td>
<td>-0.039</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>435</td>
<td>-0.170</td>
<td>0.020</td>
<td>0.065</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>301</td>
<td>-0.213</td>
<td>0.101</td>
<td>0.714</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>757</td>
<td>0.015</td>
<td>0.968</td>
<td>1.203</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>554</td>
<td>-0.112</td>
<td>0.661</td>
<td>0.984</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>393</td>
<td>-0.140</td>
<td>0.470</td>
<td>1.635</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>228</td>
<td>-0.170</td>
<td>0.012</td>
<td>2.333</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>188</td>
<td>-0.213</td>
<td>0.025</td>
<td>0.486</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>1844</td>
<td>-0.108</td>
<td>0.053</td>
<td>0.054</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>1293</td>
<td>-0.068</td>
<td>-0.025</td>
<td>-0.044</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>876</td>
<td>-0.176</td>
<td>-0.213</td>
<td>-0.065</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>544</td>
<td>-0.255</td>
<td>-0.233</td>
<td>0.656</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>325</td>
<td>-0.395</td>
<td>0.439</td>
<td>2.178</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>1640</td>
<td>-0.081</td>
<td>-0.051</td>
<td>0.014</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>1143</td>
<td>-0.068</td>
<td>0.022</td>
<td>0.089</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>790</td>
<td>-0.048</td>
<td>0.129</td>
<td>0.199</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>529</td>
<td>0.368</td>
<td>0.301</td>
<td>0.371</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>350</td>
<td>0.613</td>
<td>0.566</td>
<td>0.644</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>2028</td>
<td>-0.001</td>
<td>0.098</td>
<td>0.327</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>1526</td>
<td>0.002</td>
<td>-0.013</td>
<td>0.137</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1132</td>
<td>-0.194</td>
<td>-0.012</td>
<td>0.184</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>823</td>
<td>-0.263</td>
<td>-0.010</td>
<td>0.253</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>569</td>
<td>0.021</td>
<td>-0.070</td>
<td>-0.336</td>
</tr>
</tbody>
</table>
5.3 Inter Mode Predictors

All inter modes are considered equally computationally constraining and the average number of evaluated modes are used in order to describe the computational complexity. A complete rate distortion calculation is performed for all nine modes when the complexity is 100%. A complexity of 10% corresponds to testing approximately one mode in average. There could be complexities below 10% since skip is considered as (almost) free, but then there would be severe quality degradations.

5.3.1 Preprocessing

There is some processing of a frame performed prior to encoding. The measurement referred to as pixdiff, discussed in chapter 4.3.1, is calculated for all residual macroblocks in the frame. These values are then sorted and used in the classification of macroblocks in the predictors. Pixdiff is also used here in order to detect scene changes, which will be described in a later section.

5.3.2 Skip Predictor

The cheapest way to code a macroblock is to choose skip-mode, since no motion estimation is performed and no motion vectors and DCT coefficients need to be stored. A skip predictor should perform well and detect as many skip macroblocks as possible for sequences with no or modest motion. But it is of equal importance that the skip predictor does not destroy sequences of high detail/motion by marking macroblocks as skipped when they should not. There are several articles written about how to detect a skip macroblock, [6], [7] and [16], but they all have disadvantages by performing well for some sequences and destroying others. It is of great importance that the predictor can handle scene-changes or dramatic increases/decreases of motion in a sequence. The aim is to construct a robust and reliable skip predictor performing well for all sequences and QP:s.

To make sure that the predictor does not mark too many macroblocks as skipped in a certain frame, an estimate of the number of skip macroblocks in the frame is used as an upper limit. An estimate based on the number of skipped macroblocks in the previous frame performs satisfactory (unless there is a change of scene).
5.3. **INTER MODE PREDICTORS**

Which macroblocks in a frame should be set as skipped? According to observation (7) the probability that a macroblock is coded as skipped tends to be higher when pixdiff is low. Preprocessing have calculated and sorted the pixdiff values for all macroblocks in a frame and the macroblocks that generates the lowest pixdiff value should be classified as skip. The amount of macroblocks classified as skip varies on the demand on quality versus speed.

Note that according to observation(4) in chapter 4.2.2 the probability for skip increase if the macroblock to the left and/or above the current macroblock is coded as skipped. In the same way, observation(3) in chapter 4.2.2, tells us that probability for skip increases if the macroblock in the same position in the recent frame was coded as skipped. Therefore the knowledge of previously coded macroblocks will increase the probability for a correctly predicted macroblock. The functionality of the skip predictor is illustrated in figure 5.3.

**Results** The skip predictor has been tested with a number of sequences and quantization parameter values. The results of the skip predictor with certain settings are presented in tables 5.3 and in figure 5.4 but the results are similar with other settings. The increase in rate and change in PSNR are obtained by comparing with complexity 100%.

It is obvious that more macroblocks can be categorized as skip for large quantization parameters. Qp = 20 and Qp = 36 for Carphone have roughly the same PSNR drop and increase in rate but the complexity differs from 91% to 61%. This corresponds to evaluating in average 8.2 and 5.5 modes respectively. Comparing Coastguard and Claire at Qp = 20 shows that the amount of macroblocks that can be categorized as skip varies a lot. The skip predictor gives a complexity of 99% with Coastguard while with Claire a complexity of 40% and better PSNR and rate performance are obtained. These results corresponds well to observations (1) and (2), seen in the statistical analysis.
Figure 5.3: **Overview of skip predictor.** There are three ways of categorizing a macroblock as skip. If the macroblock in the previous frame(s) has skip as optimal mode it is more likely that the current macroblock also is classified as skip.
### 5.3. INTER MODE PREDICTORS

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>ΔRate</th>
<th>ΔPSNR</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>2.90</td>
<td>-0.16</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>5.81</td>
<td>-0.30</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4.89</td>
<td>-0.24</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6.03</td>
<td>-0.25</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>2.48</td>
<td>-0.14</td>
<td>61</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>9.30</td>
<td>-0.50</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>6.84</td>
<td>-0.29</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.87</td>
<td>-0.10</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.29</td>
<td>-0.04</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.68</td>
<td>-0.05</td>
<td>29</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>0.42</td>
<td>-0.03</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>7.98</td>
<td>-0.40</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>14.30</td>
<td>-0.52</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>21.32</td>
<td>-0.52</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>10.99</td>
<td>-0.35</td>
<td>60</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>5.09</td>
<td>-0.26</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>11.16</td>
<td>-0.41</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>7.61</td>
<td>-0.33</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>13.00</td>
<td>-0.55</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>17.01</td>
<td>-0.93</td>
<td>64</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>0.81</td>
<td>-0.06</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>0.36</td>
<td>-0.03</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>0.99</td>
<td>-0.07</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.05</td>
<td>-0.06</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>1.85</td>
<td>-0.07</td>
<td>56</td>
</tr>
</tbody>
</table>

Table 5.3: **Results for the skip predictor.** Results for various sequences and Qp for the skip predictor. The results should be regarded as either rate increase or PSNR degradation. The increase in rate and change in PSNR are obtained by comparing with complexity 100%. A positive ΔRate means that more bits is needed to represent the sequence. A negative ΔPSNR means a degradation in quality.
Figure 5.4: **Rate Distortion** curve for skip predictor for Carphone.
5.3. INTER MODE PREDICTORS

5.3.3 Skip and Mode 16x16 Predictor

This predictor is a development of the skip predictor and is not as sensitive to wrong classifications as the skip predictor. According to observation (8) it is difficult to separate skip-mode and mode 16x16 by using any block measure. When the skip predictor wrongly classifies a macroblock it is probable that mode 16x16 is the correct mode. Therefore the skip and mode 16x16 predictor evaluates skip and mode 16x16 and chooses the one that have the lowest RD cost. The functionality of this predictor is illustrated in figure 5.5.

Observation (7) says that the probability of classifying a block as skip or mode 16x16 is usually high if pixdiff is low. Since this predictor is less sensitive to wrong classifications as the skip predictor more macroblocks can be categorized. Which mode chosen in the same position in previous frame is also used to avoid incorrect classification.

To make sure that the predictor does not mark too many macroblocks as skip or mode 16x16 in a certain frame, an estimate of the number skip or mode 16x16 in the frame is used as an upper limit. An estimate based on the number of skipped macroblocks and the number of macroblocks classified as mode 16x16 in the previous frame performs satisfactory (unless there is a change of scene).

Figure 5.5: Overview of skip and mode 16x16 predictor. Skip and mode 16x16 are evaluated for the n modes with lowest pixdiff. The number of classified blocks, n, depends on the demand on speed versus quality.
CHAPTER 5. OPTIMIZATION OF MODE SELECTION

Results  The skip and mode 16x16 predictor has been tested in the same way as the skip predictor. The results are presented in table 5.4 and in figure 5.6. The increase in rate and change in PSNR are obtained by comparing with complexity 100%. It is obvious that more macroblock can be categorized as skip or mode 16x16 compared to the skip predictor. Comparing Coastguard and Claire with the same quantization parameter shows that the amount of macroblocks that can be categorized as skip or mode 16x16 varies. These results corresponds well to observations (1) and (2), seen in the statistical analysis.

The complexity still denotes the average number of modes that are evaluated. A comparison between the skip predictor and the skip or mode 16x16 predictor shows that even though the complexity is much lower for the skip and mode 16x16 predictor it gives much better PSNR and rate performances.

Figure 5.6: Rate Distortion curve for skip or mode 16x16 predictor for Coastguard.
### Table 5.4: Results for the skip or mode 16x16 predictor

Results for various sequences and Qp for the skip and mode 16x16 predictor. The results should be regarded as either rate increase or PSNR degradation. A positive ∆Rate means that more bits is needed to represent the sequence. A negative ∆PSNR means a degradation in quality.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>∆Rate</th>
<th>∆PSNR</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>2.17</td>
<td>-0.10</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2.13</td>
<td>-0.11</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>2.23</td>
<td>-0.11</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.71</td>
<td>-0.06</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.92</td>
<td>-0.07</td>
<td>28</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>0.55</td>
<td>-0.03</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>0.62</td>
<td>-0.03</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.29</td>
<td>-0.07</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.74</td>
<td>-0.01</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.11</td>
<td>-0.03</td>
<td>26</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>3.69</td>
<td>-0.26</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4.21</td>
<td>-0.22</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>3.82</td>
<td>-0.14</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4.06</td>
<td>-0.12</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.73</td>
<td>-0.02</td>
<td>28</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>7.92</td>
<td>-0.40</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>7.70</td>
<td>-0.30</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>5.53</td>
<td>-0.24</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.49</td>
<td>-0.15</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>3.31</td>
<td>-0.13</td>
<td>28</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>0.53</td>
<td>-0.04</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>0.77</td>
<td>-0.06</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.48</td>
<td>-0.13</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.86</td>
<td>-0.14</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>2.74</td>
<td>-0.15</td>
<td>40</td>
</tr>
</tbody>
</table>
5.3.4 Mode 8x8 Predictor

Previously discussed predictors have had the intention to classify macroblock of large block sizes and skip, i.e. predictors that should work well for sequences with limited motion and/or high QP - values. There is also a need of a mode 8x8 predictor, that predicts whether or not modes with large block-size, i.e. 16x16, 16x8 and 8x16 should be evaluated. If classified as mode 8x8 only the submodes are evaluated. Figure 5.7 illustrates the functionality of the mode 8x8 predictor.

![Diagram of Mode 8x8 Predictor](image)

Figure 5.7: Overview of mode 8x8 predictor. There are two ways of categorizing a macroblock as mode 8x8. If the macroblock in the previous frame has mode 8x8 as optimal mode it is more likely that the current macroblock also is classified as mode 8x8.

According to observation (7) in section 4.3, the probability of classifying a block as a mode 8x8 is high when pixdiff is high. So the macroblocks with the highest pixdiff values will be classified as mode 8x8. An estimate of the number of macroblocks classified as mode 8x8 in the frame is used as an upper limit. An estimate based on the number of macroblock classified as mode 8x8 in the previous frame performs satisfactory (unless there is a change of scene). The number of macroblock categorized as mode 8x8 by the
5.3. **INTER MODE PREDICTORS**

predictor depends on demands on speed versus quality.

The knowledge of which mode is coded in the same position in the previous frame is also used in order to avoid incorrect classification. If the macroblock in the previous frame has mode 8x8 as optimal mode it is more likely that the current macroblock also is classified as mode 8x8.

**Results** The results of the mode 8x8 predictor are presented in table 5.5 and in figure 5.8. The increase in rate and change in PSNR are obtained by comparing with complexity 100%. In contrary to the two previous predictors the mode 8x8 predictor performs better for low quantization parameter values and sequences with a lot of movement and details. The results correspond with observation (1) and (2) in the statistical analysis.

![Rate Distortion curve for mode 8x8 predictor for Foreman.](image)

Figure 5.8: **Rate Distortion** curve for mode 8x8 predictor for Foreman.
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>∆Rate</th>
<th>∆PSNR</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>3.93</td>
<td>-0.20</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>9.89</td>
<td>-0.50</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>12.72</td>
<td>-0.63</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>10.06</td>
<td>-0.40</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>7.04</td>
<td>-0.34</td>
<td>95</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>1.39</td>
<td>-0.08</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2.56</td>
<td>-0.12</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4.41</td>
<td>-0.29</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4.78</td>
<td>-0.21</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>4.25</td>
<td>-0.33</td>
<td>98</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>2.47</td>
<td>-0.18</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4.06</td>
<td>-0.22</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>3.63</td>
<td>-0.14</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.29</td>
<td>-0.04</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>-0.06</td>
<td>-0.00</td>
<td>96</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>5.38</td>
<td>-0.27</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>7.35</td>
<td>-0.30</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>9.01</td>
<td>-0.39</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5.58</td>
<td>-0.22</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>2.61</td>
<td>-0.18</td>
<td>97</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>1.47</td>
<td>-0.11</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>0.84</td>
<td>-0.06</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.92</td>
<td>-0.14</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.51</td>
<td>-0.15</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>3.58</td>
<td>-0.19</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 5.5: **Results for the mode 8x8 predictor.** Results for various sequences and QP for the mode 8x8 predictor. The results should be regarded as either rate increase or PSNR degradation. The increase in rate and change in PSNR are obtained by comparing with complexity 100%. A positive ∆Rate means that more bits is needed to represent the sequence. A negative ∆PSNR means a degradation in quality. The complexity denotes the average number of modes that are evaluated.
5.3.5 Submode Predictor

The size of the subpartitions are ranging from 8x8 pixels to 4x4 pixels. This implies that if all sub modes are evaluated, 36 motion estimations need to be performed for each macroblock in each reference frame! It is obvious that evaluating submodes is a time consuming process. In order to avoid all these calculations a submode predictor has been developed. It decides whether or not it is necessary to evaluate all submodes or if it is enough with submode 8x8. Figure 5.9 shows a flow graph of the predictor. Pixdiff for each subblock is compared with a threshold to decide whether or not the submodes 8x4, 4x8 and 4x4 should be tested. The threshold is based on pixdiff for subblocks. The threshold is calculated adaptively where old values are regarded with less importance. These adaptive calculations are explained further in Appendix A.

![Flow graph of the predictor](image)

Figure 5.9: **Overview of submode predictor.** If pixdiff for a subblock is below a threshold the subblock is classified as submode 8x8 otherwise all submodes are evaluated
Results  Table 5.6 and figure 5.10 presents the results of the subblock predictor. The increase in rate and change in PSNR are obtained by comparing with complexity 100%. The submode predictor performs better for high quantization parameter values and for sequences with limited movement and details. This corresponds with the observations from the statistical analysis.

Figure 5.10: **Rate Distortion** curve for subblock predictor for Carphone.
### 5.3. INTER MODE PREDICTORS

#### Table 5.6: Results for the submode predictor.
Results for various sequences and Qp for the submode predictor. The results should be regarded as either rate increase or PSNR degradation. A positive ΔRate means that more bits is needed to represent the sequence. A negative ΔPSNR means a degradation in quality. The complexity denotes the average number of modes that are evaluated.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>ΔRate</th>
<th>ΔPSNR</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>4.78</td>
<td>-0.24</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>3.75</td>
<td>-0.19</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>2.68</td>
<td>-0.16</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.75</td>
<td>-0.02</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.13</td>
<td>-0.06</td>
<td>47</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>3.86</td>
<td>-0.22</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>3.85</td>
<td>-0.16</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.32</td>
<td>-0.08</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.98</td>
<td>-0.06</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>-0.72</td>
<td>0.03</td>
<td>59</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>2.39</td>
<td>-0.17</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2.98</td>
<td>-0.15</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>2.36</td>
<td>-0.09</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.94</td>
<td>-0.05</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>-2.49</td>
<td>0.02</td>
<td>82</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>5.75</td>
<td>-0.29</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4.71</td>
<td>-0.19</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>2.54</td>
<td>-0.11</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>-1.04</td>
<td>0.00</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.75</td>
<td>-0.10</td>
<td>46</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>5.03</td>
<td>-0.35</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>5.76</td>
<td>-0.37</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>5.61</td>
<td>-0.41</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6.26</td>
<td>-0.37</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>4.88</td>
<td>-0.28</td>
<td>47</td>
</tr>
</tbody>
</table>
5.3.6 Rate Distortion Cost Predictor

The idea with the RD cost predictor is to use information gained from previously coded modes in a macroblock. As the name reveals it makes use of the rate distortion cost. The predictor is based on observation (11) discussed in section 4.3.2. A full evaluation of mode 16x16 is performed and the result, i.e. the RD cost, is used to classify which group of modes that is most likely to contain the best mode. There are three groups:

- Large blocks: 16x16, 16x8 and 8x16.
- Small blocks: 8x8, 8x4, 4x8 and 4x4.
- All modes including intra modes.

Figure 5.11 shows the functionality of this predictor. The thresholds are adaptively calculated using the rate distortion costs for mode 16x16 and mode 8x8 respectively. These adaptive calculations are explained in Appendix A. If the rate distortion cost for mode 16x16 is lower than $T_{16 \times 16}$ the macroblock consists of large blocks. Else if the rate distortion cost is larger than $T_{8 \times 8}$ the macroblock consists of small blocks. If the rate distortion cost does not fulfill any of these requirements all modes are tested, including intra modes.

5.3.7 Results

The rate distortion cost predictor has been tested using various sequences and QPs at a wide range of complexities. The complexity denotes the average number of modes that are evaluated. Table 5.7 presents the results for complexities around 50%. Rate and PSNR are compared to complexity 100% and the increase in rate and drop in PSNR are presented. The tables shows that the predictor performs better for low values on the quantization parameter. Figure 5.18 shows the rate distortion properties for Foreman at complexity 50%.
Figure 5.11: **Overview of the RD cost predictor.** Motion estimation is performed for mode 16x16. If rate distortion cost is below a threshold large block sizes are evaluated. Otherwise if rate distortion cost is over a threshold small block sizes are evaluated. If neither of these conditions are fulfilled all modes are evaluated.
Table 5.7: **Results for the RD cost predictor.** Results for various sequences and Qp for the rate distortion cost predictor. The results should be regarded as either rate increase or PSNR degradation. A positive $\Delta$Rate means that more bits is needed to represent the sequence. A negative $\Delta$PSNR means a degradation in quality. The complexity denotes the average number of modes that are evaluated.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>$\Delta$Rate</th>
<th>$\Delta$PSNR</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>3.29</td>
<td>-0.17</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4.80</td>
<td>-0.25</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>6.51</td>
<td>-0.33</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>12.32</td>
<td>-0.54</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>14.77</td>
<td>-0.84</td>
<td>54</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>1.79</td>
<td>-0.10</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2.34</td>
<td>-0.11</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4.48</td>
<td>-0.30</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>9.51</td>
<td>-0.40</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>11.68</td>
<td>-0.78</td>
<td>52</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>1.51</td>
<td>-0.10</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2.26</td>
<td>-0.12</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4.63</td>
<td>-0.19</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>10.91</td>
<td>-0.27</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>15.10</td>
<td>-0.41</td>
<td>53</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>4.40</td>
<td>-0.22</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>8.18</td>
<td>-0.32</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>11.14</td>
<td>-0.45</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>9.58</td>
<td>-0.45</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>10.54</td>
<td>-0.58</td>
<td>55</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>0.74</td>
<td>-0.05</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>0.86</td>
<td>-0.05</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>1.16</td>
<td>-0.10</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.85</td>
<td>-0.12</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>3.39</td>
<td>-0.17</td>
<td>58</td>
</tr>
</tbody>
</table>
Figure 5.12: **Rate Distortion** curve for RD cost predictor for Foreman.
5.4 Reference Frame Predictor

Multiple reference frames in H.264 is discussed in chapter 3.4.1. The use of several reference frames affect the computational complexity of motion estimation in a linear way. Therefore it is necessary to limit the maximum number of reference frames used. According to observation (5) the best match, i.e. the lowest motion estimation cost, is often found in frames that are close to the current frame. The need of multiple reference frames in order to get a good match is highly depended on the sequence. Therefore a simple limitation of the number of reference frames would not be wise because of the wide spectrum of movie material available.

A flowchart of the predictor can be found in figure 5.13. The basic idea with the predictor is to use the results from the motion estimation for mode 16x16 to limit the number of reference frames for modes 16x8 and 8x16. The same idea is utilized for the submodes. A full motion estimation is performed for mode 8x8 in the first subblock and the optimal reference frame is used to limit the number of reference frames in motion estimation for other submodes.

If motion estimation in a reference frame for mode 16x16 yields a high motion estimation cost, this particular frame is not used in motion estimation for modes 16x8 and 8x16. The motion estimation cost for mode 8x8 is used analogously in order to skip reference frames in the motion estimation for the other submodes.

Results The two rightmost columns in table 5.8 presents results from the reference frame predictor. The predictor has been set to test about 2 reference frames in average. PSNR and rate obtained without using the predictor is also presented in table 5.8. A comparison shows that the results with the reference frame predictor using 2 reference frames in average is almost as good as using 10 reference frames without applying the predictor. Since the time consumption is linear to the number of reference frames the motion estimation should have become five times faster using the predictor, without loosing noticeable image quality.
5.4. REFERENCE FRAME PREDICTOR

Figure 5.13: Overview of reference predictor.
### Table 5.8: Reference frames

PSNR and rate (kbit/s) for various test sequences with quantization parameter 28. The rate and PSNR for 10 reference frames is used as reference. A positive $\Delta$Rate means that more bits is needed to represent the sequence. A negative $\Delta$PSNR means a degradation in quality.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Rate</th>
<th>PSNR (dB)</th>
<th>$\Delta$Rate</th>
<th>$\Delta$PSNR</th>
<th>$\Delta$Rate</th>
<th>$\Delta$PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>95.44</td>
<td>37.60</td>
<td>9.12</td>
<td>-0.16</td>
<td>1.84</td>
<td>-0.08</td>
</tr>
<tr>
<td>Claire</td>
<td>31.30</td>
<td>39.74</td>
<td>1.30</td>
<td>-0.02</td>
<td>0.22</td>
<td>-0.01</td>
</tr>
<tr>
<td>Coastguard</td>
<td>214.44</td>
<td>34.37</td>
<td>-0.70</td>
<td>0.02</td>
<td>-1.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Foreman</td>
<td>113.79</td>
<td>36.20</td>
<td>1.46</td>
<td>-0.06</td>
<td>0.18</td>
<td>-0.02</td>
</tr>
<tr>
<td>Scenecut</td>
<td>51.80</td>
<td>33.63</td>
<td>13.19</td>
<td>-0.49</td>
<td>0.42</td>
<td>0.01</td>
</tr>
</tbody>
</table>
5.5 Scene Change Detection

Modern video sequences (like music videos) often contain numerous scene changes. The probability of motion estimation finding a decent match in a reference frame from another scene is rather small. Therefore it can be more efficient to use intra modes when a scene change is detected. There is also a question about what should be considered as a scene change. If two consecutive frames belong to complete different scenes, it's obvious that a scene change should be detected. Should a partially changed frame also be considered as a scene change? If not, there would be areas in the current frame that would not find a decent match through motion estimation. To avoid those poor matches, the threshold is adapted so the intra modes are evaluated as well on these partially changed scenes.

Since preprocessing calculates pixdiff for the entire frame and because it is based on the difference between the current and the first reference frame, it is used to detect scene changes. A scene change is detected when the mean value of pixdiff is larger than a threshold. In case of a scene change intra modes are evaluated, in addition to the other modes. Figure 5.14 shows a flowgraph over the scene change detection algorithm.

5.5.1 Results

The movie sequence Scenecut contains numerous scene changes, both complete and partially. With an appropriate value on the threshold, they are all detected and the intra modes are switched on. The only drawback with the detection of partially changed scenes is that more scene changes than necessary might be detected. The only harm done is that the intra modes are evaluated as well, resulting in a tiny increase in encoding time. There is however no possibility for quality degradation.
Figure 5.14: **Overview of the scene change predictor.** A scene change is detected, and intra modes are tested, if the absolute mean value of pixdiff is larger than a threshold.
5.6 Regulator

The aim with this thesis has been to develop as general predictors as possible, i.e. they should perform well independent of sequence and quantization parameter. In order to achieve this, determining parameters for the predictors becomes quite complicated.

By alternating the parameters the complexity can be regulated. A regulator has been developed in order to set parameters to obtain a predefined complexity. An overview of the regulator can be found in figure 5.15. It shows that a target complexity, \( c_T(t) \), is used as input. The complexity, \( c(t) \) for the previous frame(s) is fed back and is subtracted from the target complexity signal to form a error signal, \( e(t) \). The complexity target should of course be possible to alter online.

The easiest regulator is a P - regulator, in which the output, \( u(t) \), is proportional to the error signal, \( e(t) \). The usage of a P - regulator affected the stability of the system negatively. In order to manage this problem another part is added, where the output is proportional to the derivate of the error signal.

A regulator consisting of both a proportional and a derivating part is called a PD - regulator. The output, \( u(t) \), is formed according to equation 5.1.

\[
    u(t) = K_p \left( e(t) + T_d \frac{d(e(t))}{dt} \right)
\]

where \( K_p \) and \( T_d \) are constants.

Figure 5.15: Overview of the regulator The complexity, \( c \), obtained by the predictors is subtracted from the target complexity \( c_T \). This difference is feed to the regulator and the output is used to regulate the predictors.
5.6.1 Results

Figure 5.16 shows the results of the regulator. The target complexity is changed during the encoding and the complexity follows as supposed. The settling time is fairly short, i.e. the time for the complexity to have settled around the target complexity. A settling time of ten frames when the complexity target is changed from 75% to 15%, might seem like a long time but it is just 1/3 seconds with a framerate at 30 fps.

Figure 5.16: Results of the regulator The complexity target, solid line, have been altered from 75% to 15% and then to 35%. The predictors are regulated in order for the complexity, dashed line, to follow the target.
5.7 Mode Selection Predictor

Figure 5.17 shows a flowchart over how predictors are combined to achieve a complete mode selection algorithm. A summary of which modes are evaluated for each predictor is presented in table 5.7. The first predictor finds blocks that could be categorized as skip. The skip predictor is followed by a predictor that evaluates skip and mode 16x16. Then the mode 8x8 predictor finds subblock that can be categorized as either submodes 8x8, 8x4, 4x8 or 4x4. The rate distortion predictor classifies macroblock that are not categorized by neither of the predictors above. The sub block predictor is applied when the sub modes are to be tested. It decides whether or not submodes 8x4, 4x8 and 4x4 are necessary.

Some properties of the mode selection predictor should be further explained. With a complexity target at 100 %, all macroblocks fall into branch F, where all modes are evaluated. Decreasing the complexity target to 80-90 %, most of the macroblocks will still end up in branch F, while some macroblocks starts to fall into other branches like E, G, H and I. Decreasing the complexity target further down to around 40-50 %, macroblocks will be rather evenly distributed in the different branches. If the complexity target is set to 10-20 %, e.g. there is only time to evaluate a small fraction of the modes, then most macroblocks will fall into branches A, B and C. The distribution of macroblocks in branches B and C, both rather cheap, is dependent of the quantization parameter as well as the amount of motion in the sequence. If the quantization parameter is large and the movie sequence contain low motion, most macroblocks will end up in branch B, that is forced into mode skip or mode 16x16. If, on the other hand, the quantization parameter is small and the movie sequence contain high detail/motion, most of the macroblocks will end up in branch C and be categorized as mode 8x8.

At complexity targets below 10 %, macroblocks are forced to be skipped and therefore end up in branch A. Complexity targets below 10 % should only be used in extreme conditions since the quality can decrease dramatically.
CHAPTER 5. OPTIMIZATION OF MODE SELECTION

Figure 5.17: Overview of the combined predictor.
### Table 5.9: Summary of which modes are evaluated by the predictors

<table>
<thead>
<tr>
<th>Exit</th>
<th>Modes evaluated</th>
<th>Number of modes evaluated</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(Skip)</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>16x16</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>8x8</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>8x8, 8x4, 4x8, 4x4</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>16x16, 16x8, 8x16</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>16x16, 16x8, 8x16, 8x8</td>
<td>4</td>
</tr>
<tr>
<td>G</td>
<td>All modes, including intra</td>
<td>9</td>
</tr>
<tr>
<td>H</td>
<td>16x16, 8x8</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>16x16, 8x8, 8x4, 4x8, 4x4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.9: **Summary** of which modes are evaluated by the predictors
5.7.1 Results

The mode selection predictor has been tested using various sequences and QPs at a wide range of complexities. The complexity denotes the average number of modes that are evaluated. Tables 5.10 and 5.11 presents the rate and PSNR for various complexities. The tables shows that the mode selection algorithm performs very well. At complexities over 50% there is almost no quality degradation. The limit for perceptual image quality degradation, 0.5 dB, is exceeded at complexity 10% for some QPs. This complexity corresponds to evaluating only one mode in average, which will make encoding very fast. Figure 5.18 shows the great rate distortion properties for Scenecut at complexity 25%.

![Rate Distortion curve for the mode selection predictor for Scenecut.](image)

Figure 5.18: **Rate Distortion** curve for the mode selection predictor for Scenecut.
### 5.7. MODE SELECTION PREDICTOR

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>Rate (kbit/s)</th>
<th>ΔRate</th>
<th>ΔRate</th>
<th>ΔRate</th>
<th>ΔRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>284.52</td>
<td>0.97</td>
<td>1.43</td>
<td>4.62</td>
<td>7.46</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>163.07</td>
<td>0.41</td>
<td>1.58</td>
<td>6.73</td>
<td>9.27</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>92.43</td>
<td>1.61</td>
<td>1.85</td>
<td>7.79</td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>52.26</td>
<td>0.25</td>
<td>2.33</td>
<td>9.76</td>
<td>16.41</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>31.15</td>
<td>1.37</td>
<td>1.49</td>
<td>12.31</td>
<td>20.25</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>98.15</td>
<td>0.16</td>
<td>0.98</td>
<td>5.74</td>
<td>10.80</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>55.39</td>
<td>0.12</td>
<td>1.27</td>
<td>6.62</td>
<td>14.58</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>30.91</td>
<td>0.60</td>
<td>2.05</td>
<td>7.19</td>
<td>11.55</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>18.07</td>
<td>-0.07</td>
<td>1.83</td>
<td>6.86</td>
<td>18.25</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>11.51</td>
<td>1.25</td>
<td>0.79</td>
<td>4.97</td>
<td>14.57</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>789.23</td>
<td>0.60</td>
<td>1.21</td>
<td>2.39</td>
<td>3.49</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>467.25</td>
<td>0.51</td>
<td>1.14</td>
<td>3.24</td>
<td>5.27</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>243.84</td>
<td>0.83</td>
<td>2.02</td>
<td>6.36</td>
<td>11.12</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>107.79</td>
<td>0.10</td>
<td>1.79</td>
<td>12.92</td>
<td>18.18</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>47.78</td>
<td>1.27</td>
<td>5.03</td>
<td>16.02</td>
<td>18.47</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>396.40</td>
<td>1.09</td>
<td>2.03</td>
<td>4.56</td>
<td>7.92</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>207.07</td>
<td>1.68</td>
<td>2.57</td>
<td>6.82</td>
<td>12.10</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>107.94</td>
<td>0.73</td>
<td>2.19</td>
<td>10.13</td>
<td>17.63</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>60.11</td>
<td>0.32</td>
<td>-0.29</td>
<td>10.71</td>
<td>18.51</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>37.02</td>
<td>-0.25</td>
<td>0.08</td>
<td>7.08</td>
<td>17.52</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>772.71</td>
<td>0.91</td>
<td>1.36</td>
<td>2.23</td>
<td>3.32</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>448.56</td>
<td>0.61</td>
<td>0.90</td>
<td>2.00</td>
<td>3.48</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>329.66</td>
<td>0.60</td>
<td>1.06</td>
<td>2.20</td>
<td>3.83</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>216.70</td>
<td>0.09</td>
<td>0.84</td>
<td>2.87</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>133.98</td>
<td>0.51</td>
<td>1.01</td>
<td>4.05</td>
<td>6.72</td>
</tr>
</tbody>
</table>

Table 5.10: **Results for the mode selection predictor.** Results for various sequences and Qp for the mode selection predictor. The four right columns represent the increase in rate for complexity 75%, 50%, 25% and 10% respectively. Complexity 100% is used as reference for rate comparison. A negative ΔRate means a degradation in quality.
### Table 5.11: Results for the mode selection predictor

Results for various sequences and Qp for the mode selection predictor. The four right columns represent the change in PSNR for complexity 75%, 50%, 25% and 10% respectively. Complexity 100% is used as reference for PSNR comparison. A negative ∆PSNR means a degradation in quality.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>PSNR (dB)</th>
<th>∆PSNR</th>
<th>∆PSNR</th>
<th>∆PSNR</th>
<th>∆PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>43.08</td>
<td>-0.05</td>
<td>-0.08</td>
<td>-0.24</td>
<td>-0.38</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>40.21</td>
<td>-0.03</td>
<td>-0.07</td>
<td>-0.32</td>
<td>-0.44</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>37.39</td>
<td>-0.09</td>
<td>-0.09</td>
<td>-0.38</td>
<td>-0.72</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>34.54</td>
<td>-0.02</td>
<td>-0.11</td>
<td>-0.45</td>
<td>-0.73</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>31.98</td>
<td>-0.06</td>
<td>-0.05</td>
<td>-0.62</td>
<td>-1.00</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>45.41</td>
<td>-0.01</td>
<td>-0.05</td>
<td>-0.30</td>
<td>-0.54</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>42.63</td>
<td>-0.01</td>
<td>-0.06</td>
<td>-0.26</td>
<td>-0.57</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>39.77</td>
<td>-0.05</td>
<td>-0.13</td>
<td>-0.39</td>
<td>-0.65</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>36.91</td>
<td>0.01</td>
<td>-0.04</td>
<td>-0.27</td>
<td>-0.72</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>34.57</td>
<td>-0.01</td>
<td>-0.10</td>
<td>-0.38</td>
<td>-0.91</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>40.66</td>
<td>-0.04</td>
<td>-0.09</td>
<td>-0.18</td>
<td>-0.26</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>37.19</td>
<td>-0.02</td>
<td>-0.07</td>
<td>-0.18</td>
<td>-0.30</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>34.04</td>
<td>-0.03</td>
<td>-0.11</td>
<td>-0.27</td>
<td>-0.48</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>34.04</td>
<td>-0.02</td>
<td>-0.06</td>
<td>-0.34</td>
<td>-0.47</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>31.02</td>
<td>-0.06</td>
<td>-0.11</td>
<td>-0.57</td>
<td>-0.56</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>41.56</td>
<td>-0.05</td>
<td>-0.11</td>
<td>-0.22</td>
<td>-0.38</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>38.49</td>
<td>-0.06</td>
<td>-0.11</td>
<td>-0.28</td>
<td>-0.49</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>35.78</td>
<td>-0.04</td>
<td>-0.10</td>
<td>-0.42</td>
<td>-0.71</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33.19</td>
<td>-0.00</td>
<td>0.00</td>
<td>-0.46</td>
<td>-0.76</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>30.75</td>
<td>0.08</td>
<td>-0.07</td>
<td>-0.43</td>
<td>-0.85</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>42.93</td>
<td>-0.08</td>
<td>-0.11</td>
<td>-0.17</td>
<td>-0.26</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>39.86</td>
<td>-0.03</td>
<td>-0.07</td>
<td>-0.15</td>
<td>-0.25</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>36.93</td>
<td>-0.04</td>
<td>-0.07</td>
<td>-0.19</td>
<td>-0.29</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33.73</td>
<td>-0.02</td>
<td>-0.05</td>
<td>-0.20</td>
<td>-0.31</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>30.60</td>
<td>-0.01</td>
<td>-0.07</td>
<td>-0.23</td>
<td>-0.41</td>
</tr>
</tbody>
</table>
Chapter 6
Proposed Algorithm

The final proposed algorithm consists of four parts.

1. **Mode Selection Predictor** This predictor consists of a handful specialized predictors and constitutes the central part of the algorithm. The purpose of the predictor is to reduce the number of modes to be tested. This is described in detail in section 5.7.

2. **Reference Frame Predictor** This predictor, described in section 5.4, can have significant importance in time saving, since time spent in motion estimation is approximately proportional to the number of reference frames evaluated.

3. **Scene Change Detection** Some types of video often contain frequent occurring scene cuts and it is of importance to handle these in a clever way. The scene change detection is further described in section 5.5.

4. **Regulator** The regulator’s assignment is to keep the encoding at a certain level. Encoding at a high complexity target will be time consuming, but good quality will be obtained. A low level will degrade the quality but speed up the encoding. The level can be set in the beginning of the encoding and be changed during encoding, e.g. due to available computational power. The regulator is further described in section 5.6.

Previously the complexity has been defined as the average number of modes that are evaluated. In the proposed algorithm the complexity also
includes the number of reference frames evaluated. The relationship between the complexity and number of reference frames is not linear as for the modes. The relationship is rather adapted to obtain optimal tradeoff between quality and time consumption. For example a complexity of 50% corresponds to 2.2 reference frames in average while 25% corresponds to 1.9 reference frames in average.

In order to measure how the proposed algorithm actually affect the coding time it has been implemented in Popwire’s H.264 codec. The results presented below are obtained by using this codec on an Apple G5 dual 2.0 GHz.

### 6.1 Results

The proposed algorithm has been tested using various sequences, quantization parameters and complexities. PNSR, rate and coding time are represented in tables 6.1, 6.2 and 6.3 respectively. The coding time is stated in fps, i.e. how many frames that are encoded each second. By comparing these tables the tradeoff between quality and speed is obvious. At complexities over 50% the PSNR drop is within the limit for perceptual image quality degradation. At complexity 50% the encoding becomes 6 times faster which is sufficient for real time encoding. For complexities 25% and 10% there are noticeable quality degradations but the coding becomes very fast. Complexity 10% corresponds to about 17 times faster encoding.
### 6.1. RESULTS

Table 6.1: **Results for the proposed algorithm** Results for various sequences and Qp for the proposed algorithm. The four right columns represent the change in PSNR for complexity 75%, 50%, 25% and 10% respectively. Complexity 100% is used as reference for PSNR comparison. A negative ΔPSNR means a degradation in quality.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>PSNR(dB)</th>
<th>ΔPSNR 75%</th>
<th>ΔPSNR 50%</th>
<th>ΔPSNR 25%</th>
<th>ΔPSNR 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>42.94</td>
<td>-0.22</td>
<td>-0.53</td>
<td>-0.89</td>
<td>-1.89</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>40.05</td>
<td>-0.19</td>
<td>-0.51</td>
<td>-0.87</td>
<td>-1.91</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>37.21</td>
<td>-0.15</td>
<td>-0.43</td>
<td>-0.77</td>
<td>-1.71</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>34.25</td>
<td>-0.07</td>
<td>-0.34</td>
<td>-0.67</td>
<td>-1.61</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>31.43</td>
<td>-0.05</td>
<td>-0.28</td>
<td>-0.50</td>
<td>-1.27</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>45.40</td>
<td>-0.13</td>
<td>-0.26</td>
<td>-0.60</td>
<td>-1.15</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>42.64</td>
<td>-0.09</td>
<td>-0.23</td>
<td>-0.55</td>
<td>-1.08</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>39.80</td>
<td>-0.08</td>
<td>-0.19</td>
<td>-0.39</td>
<td>-0.88</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>36.87</td>
<td>-0.08</td>
<td>-0.17</td>
<td>-0.36</td>
<td>-0.71</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>34.28</td>
<td>0.04</td>
<td>-0.02</td>
<td>-0.16</td>
<td>-0.49</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>40.88</td>
<td>-0.12</td>
<td>-0.34</td>
<td>-0.62</td>
<td>-0.94</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>37.56</td>
<td>-0.06</td>
<td>-0.26</td>
<td>-0.52</td>
<td>-0.90</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>34.58</td>
<td>-0.06</td>
<td>-0.20</td>
<td>-0.42</td>
<td>-0.78</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>31.64</td>
<td>-0.04</td>
<td>-0.14</td>
<td>-0.35</td>
<td>-0.68</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>29.01</td>
<td>-0.04</td>
<td>-0.04</td>
<td>-0.32</td>
<td>-0.62</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>41.90</td>
<td>-0.25</td>
<td>-0.61</td>
<td>-1.02</td>
<td>-1.77</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>38.94</td>
<td>-0.19</td>
<td>-0.48</td>
<td>-0.87</td>
<td>-1.61</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>36.33</td>
<td>-0.18</td>
<td>-0.46</td>
<td>-0.80</td>
<td>-1.51</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33.57</td>
<td>-0.12</td>
<td>-0.35</td>
<td>-0.74</td>
<td>-1.43</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>30.87</td>
<td>-0.03</td>
<td>-0.22</td>
<td>-0.55</td>
<td>-1.25</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>44.19</td>
<td>-0.27</td>
<td>-0.34</td>
<td>-0.83</td>
<td>-1.48</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>40.84</td>
<td>-0.36</td>
<td>-0.52</td>
<td>-1.38</td>
<td>-1.76</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>36.85</td>
<td>-0.48</td>
<td>-0.23</td>
<td>-1.22</td>
<td>-1.91</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33.36</td>
<td>-0.17</td>
<td>-0.00</td>
<td>-0.75</td>
<td>-1.53</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>30.26</td>
<td>0.17</td>
<td>0.03</td>
<td>-0.24</td>
<td>-0.94</td>
</tr>
</tbody>
</table>
### Table 6.2: Results for the Proposed Algorithm

Results for various sequences and Qp for the proposed algorithm. The four right columns represent the change in rate for complexity 75%, 50%, 25% and 10% respectively. Complexity 100% is used as reference for rate comparison. A positive Δrate means a degradation in quality.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>100% rate(kbit/s)</th>
<th>Δrate</th>
<th>75% Δrate</th>
<th>50% Δrate</th>
<th>25% Δrate</th>
<th>10% Δrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>365.63</td>
<td>3.70</td>
<td>9.18</td>
<td>16.08</td>
<td>36.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>227.65</td>
<td>3.14</td>
<td>9.10</td>
<td>16.43</td>
<td>38.72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>135.99</td>
<td>2.58</td>
<td>8.28</td>
<td>15.02</td>
<td>37.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>77.40</td>
<td>1.47</td>
<td>7.70</td>
<td>14.49</td>
<td>38.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>42.68</td>
<td>1.04</td>
<td>5.63</td>
<td>11.06</td>
<td>29.11</td>
<td></td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>88.48</td>
<td>2.85</td>
<td>6.00</td>
<td>13.38</td>
<td>27.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>51.53</td>
<td>1.79</td>
<td>4.75</td>
<td>11.66</td>
<td>24.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>29.09</td>
<td>1.91</td>
<td>3.75</td>
<td>8.66</td>
<td>18.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>16.41</td>
<td>1.70</td>
<td>3.78</td>
<td>8.12</td>
<td>17.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>9.59</td>
<td>-0.96</td>
<td>1.38</td>
<td>2.31</td>
<td>10.04</td>
<td></td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>595.90</td>
<td>1.74</td>
<td>4.90</td>
<td>9.20</td>
<td>14.28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>356.33</td>
<td>1.30</td>
<td>4.76</td>
<td>9.30</td>
<td>16.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>198.66</td>
<td>1.22</td>
<td>4.42</td>
<td>9.45</td>
<td>18.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>98.65</td>
<td>1.63</td>
<td>3.19</td>
<td>9.54</td>
<td>18.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>47.37</td>
<td>0.51</td>
<td>1.67</td>
<td>9.92</td>
<td>28.29</td>
<td></td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>421.58</td>
<td>4.56</td>
<td>11.71</td>
<td>19.96</td>
<td>36.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>242.04</td>
<td>3.85</td>
<td>10.50</td>
<td>19.54</td>
<td>38.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>137.21</td>
<td>3.68</td>
<td>10.25</td>
<td>18.24</td>
<td>37.78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>78.63</td>
<td>2.57</td>
<td>8.17</td>
<td>15.20</td>
<td>33.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>46.60</td>
<td>0.93</td>
<td>4.43</td>
<td>11.07</td>
<td>25.80</td>
<td></td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>411.40</td>
<td>4.05</td>
<td>5.36</td>
<td>13.33</td>
<td>24.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>261.70</td>
<td>4.35</td>
<td>6.27</td>
<td>16.67</td>
<td>23.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>178.23</td>
<td>4.89</td>
<td>2.26</td>
<td>13.55</td>
<td>20.58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>119.13</td>
<td>2.38</td>
<td>0.24</td>
<td>9.97</td>
<td>20.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>76.26</td>
<td>-2.58</td>
<td>-0.75</td>
<td>3.70</td>
<td>15.85</td>
<td></td>
</tr>
</tbody>
</table>
### Table 6.3: Results for the proposed algorithm

Results for various sequences and Qp for the proposed algorithm. The coding time, stated in fps, are presented for complexity 100%, 75%, 50%, 25% and 10%.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Qp</th>
<th>100% fps</th>
<th>75% fps</th>
<th>50% fps</th>
<th>25% fps</th>
<th>10% fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>20</td>
<td>5</td>
<td>18</td>
<td>28</td>
<td>46</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>5</td>
<td>18</td>
<td>28</td>
<td>48</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>5</td>
<td>18</td>
<td>29</td>
<td>49</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5</td>
<td>19</td>
<td>30</td>
<td>51</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>5</td>
<td>20</td>
<td>30</td>
<td>52</td>
<td>85</td>
</tr>
<tr>
<td>Claire</td>
<td>20</td>
<td>5</td>
<td>21</td>
<td>33</td>
<td>55</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>5</td>
<td>22</td>
<td>33</td>
<td>55</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>5</td>
<td>22</td>
<td>32</td>
<td>55</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6</td>
<td>22</td>
<td>33</td>
<td>55</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>6</td>
<td>23</td>
<td>32</td>
<td>55</td>
<td>92</td>
</tr>
<tr>
<td>Coastguard</td>
<td>20</td>
<td>4</td>
<td>18</td>
<td>28</td>
<td>44</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4</td>
<td>19</td>
<td>29</td>
<td>45</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4</td>
<td>19</td>
<td>30</td>
<td>48</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4</td>
<td>20</td>
<td>30</td>
<td>50</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>5</td>
<td>20</td>
<td>31</td>
<td>52</td>
<td>81</td>
</tr>
<tr>
<td>Foreman</td>
<td>20</td>
<td>4</td>
<td>17</td>
<td>27</td>
<td>45</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4</td>
<td>18</td>
<td>29</td>
<td>48</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>4</td>
<td>19</td>
<td>30</td>
<td>50</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4</td>
<td>19</td>
<td>30</td>
<td>52</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>5</td>
<td>20</td>
<td>31</td>
<td>53</td>
<td>84</td>
</tr>
<tr>
<td>Scenecut</td>
<td>20</td>
<td>6</td>
<td>21</td>
<td>31</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>6</td>
<td>22</td>
<td>32</td>
<td>51</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>6</td>
<td>22</td>
<td>32</td>
<td>53</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6</td>
<td>23</td>
<td>33</td>
<td>54</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>6</td>
<td>23</td>
<td>33</td>
<td>55</td>
<td>88</td>
</tr>
</tbody>
</table>
Figure 6.1: **Rate Distortion** curve for the proposed algorithm for Coastguard at complexity 50%.

Figure 6.2: **Rate Distortion** curve for the proposed algorithm for Coastguard at complexity 25%.
6.2 Conclusion/Discussion

Since video coding is quite complex the extensive statistical analysis made an excellent ground to develop the algorithm from. Since H.264 and this thesis treat a lot of video coding aspects the work have given us great knowledge in this interesting field.

The main advantage with the proposed algorithm is the fact that it performs very well for various values on the quantizations parameter and for different sequences. The predictors do not introduce any dramatic quality degradation like dips in PSNR. This is mainly because the proposed algorithm is built from a number of modules that are suitable for different conditions. Thresholds in the modules are updated adaptively which is another reason why the algorithm is robust. Since the proposed algorithm consists of a number of modules that are independent, a module could easily be exchanged if needed. The proposed algorithm is basically not codec specific but the modules could be weighted in order to be optimal for a specific codec.

A disadvantage with the regulator is that it can be a bit oscillating at high complexities. Some frames are easy to classify with modules which corresponds to very low complexities which gives a complexity drop.

Another disadvantage, or actually a trade off, is the fact that the calculations of the measurement pixdiff is more computationally constraining then the classic SAD. Since the results are better with pixdiff this measurement has been used anyway. These calculation are not very complicated and could also be optimized for different platforms.
Chapter 7

Future Work

There are some areas that could be explored further in order to complete the algorithm proposed in this thesis.

• Since the proposed algorithm is based on the use of the H.264 baseline profile there should be some adjustments and development in order to make it usable for the other profiles as well. B - frames make a specifically interesting case due to the prediction in future frames.

• Use the long term reference frame buffer in order to take advantage of previous scene changes. It is common that two different scenes are alternated and therefore it could be a good idea to store the last frame before a scene change.

• Examine if pixdiff can be further used during encoding, for example in fading detection or estimating DCT coefficients.

• Develop an algorithm for distinguish vertical and horizontal movements in order to predict which of the modes 16x8 and 8x16 is most probable. The algorithm should also be able to use to distinguish the submodes 8x4 and 4x8.
# Table of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVC</td>
<td>Advanced Video Coding</td>
</tr>
<tr>
<td>CABAC</td>
<td>Context - Based Adaptive Binary Arithmetic Coding</td>
</tr>
<tr>
<td>CAVLC</td>
<td>Context - Based Adaptive Variable Length Coding</td>
</tr>
<tr>
<td>CODEC</td>
<td>COder/DECoder</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disc</td>
</tr>
<tr>
<td>FPS</td>
<td>Frames Per Second</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>HDTV</td>
<td>High Definition Television</td>
</tr>
<tr>
<td>IDCT</td>
<td>Inverse Discrete Cosine Transform</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Pictures Expert Group</td>
</tr>
<tr>
<td>JVT</td>
<td>Joint Video Team</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>MB</td>
<td>Macroblock</td>
</tr>
<tr>
<td>ME</td>
<td>Motion Estimation</td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Pictures Expert Group</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MV</td>
<td>Motion Vector</td>
</tr>
<tr>
<td>QCIF</td>
<td>Quarter Common Intermediate Format</td>
</tr>
<tr>
<td>QP</td>
<td>Quantization Parameter</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>RD</td>
<td>Rate Distortion</td>
</tr>
<tr>
<td>RGB</td>
<td>Red-Green-Blue. Colorspace</td>
</tr>
<tr>
<td>SAD</td>
<td>Sum of Absolute Differences</td>
</tr>
<tr>
<td>VLC</td>
<td>Variable Length Code</td>
</tr>
<tr>
<td>YUV</td>
<td>Luminance-Chrominance(blue)-Chrominance(red). Colorspace</td>
</tr>
</tbody>
</table>
Bibliography


Appendix A

Mathematical Formulas

Update of Mean Value

Given a set of numbers, \( \{x_i\}_{i=1}^{N} \), the mean value is defined as

\[
m_N = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (A.1)
\]

When expanding the set to \( \{x_i\}_{i=1}^{N+1} \) the new mean value is calculated as

\[
m_{N+1} = \frac{1}{N+1} \sum_{i=1}^{N+1} x_i = \frac{1}{N+1} \left( \sum_{i=1}^{N} x_i + \frac{1}{N+1} x_{N+1} \right) = \frac{N}{N+1} m_N + \frac{1}{N+1} x_{N+1} \quad (A.2)
\]

Introducing the "forgetting factor" \( \lambda \) will give recent \( x_i \)'s larger weights than older values. Thus, the mean value will "forget" older values. This yields the following equation including \( \lambda \):

\[
m_{N+1} = \frac{N \lambda m_N + x_{N+1}}{N \lambda + 1} \quad (A.3)
\]
APPENDIX A. MATHEMATICAL FORMULAS

Update of Variance

Given a set of numbers, \( \{x_i\}_{i=1}^N \) and the corresponding mean value \( m_N \), the variance is defined as

\[
\sigma_N^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - m_N)^2 \quad (A.4)
\]

When expanding the set to \( \{x_i\}_{i=1}^{N+1} \) the new variance is calculated as

\[
\sigma_{N+1}^2 = \frac{1}{(N+1)} \sum_{i=1}^{N+1} (x_i - m_{N+1})^2 = \frac{1}{(N+1)} \left[ \sum_{i=1}^{N} (x_i - m_{N+1})^2 + (x_{N+1} - m_{N+1})^2 \right] \quad (A.5)
\]

Developing the inner sum yields

\[
\sum_{i=1}^{N} (x_i - m_{N+1})^2 = \sum_{i=1}^{N} \left( x_i^2 - 2m_{N+1}x_i + m_{N+1}^2 \right) = \\
\sum_{i=1}^{N} (x_i^2 - 2m_{N+1}x_i + m_{N+1}^2) - \sum_{i=1}^{N} 2m_{N+1}x_i + \sum_{i=1}^{N} m_{N+1}^2 = \\
\sum_{i=1}^{N} (x_i - m_N)^2 + 2(m_N - m_{N+1}) \sum_{i=1}^{N} x_i + N \left( m_{N+1}^2 - m_N^2 \right) = \\
N \sigma_N^2 + 2(m_N - m_{N+1}) \sum_{i=1}^{N} x_i + N \left( m_{N+1}^2 - m_N^2 \right)
\]

Noting that \( \sum_{i=1}^{N} x_i = Nm_N \) gives

\[
\sum_{i=1}^{N} (x_i - m_{N+1})^2 = N \sigma_N^2 + 2Nm_N(m_N - m_{N+1}) + N(m_{N+1}^2 - m_N^2) = \\
N \left( \sigma_N^2 + 2m_N^2 - 2m_Nm_{N+1} + m_{N+1}^2 - m_N^2 \right) = N \left( \sigma_N^2 + m_{N+1}^2 - 2m_Nm_{N+1} + m_{N+1}^2 \right) = \\
N \left( \sigma_N^2 + (m_N - m_{N+1})^2 \right)
\]

Insertion in equation (A.5) gives the final equation
\[ \sigma_{N+1}^2 = \frac{N}{N+1} \left[ \sigma_N^2 + (m_N - m_{N+1})^2 + \frac{(x_{i+1} - m_{N+1})^2}{N} \right] \] (A.6)

The procedure above is summarized in the following algorithm:

1. calculate the new mean value \( m_{N+1} \) according to equation (A.2) or (A.3)
2. calculate the new variance according to equation (A.6)
Appendix B

Calculation of PSNR Drop and Increase of Rate

Rate-Distortion Curves

When encoding a movie sequence for a fixed quantization parameter (Qp), one obtains a pair of values, namely the rate of the encoded sequence and the corresponding PSNR value. Doing this for a range of Qp:s yields a number of rate- and PSNR-values from which the characteristic Rate-Distortion curve can be plotted.

To evaluate if a specific change in the encoder (choosing different algorithms etc.) has large influence on the compression performance one can compare the Rate-Distortion curves. If one curve A is above one other curve B, this means that the encoder corresponding to curve A is performing better than the encoder corresponding to curve B. It is therefore easy to qualitatively compare different encoders.

To get a more quantitative comparison there are two different approaches. The first is to look at the difference in PSNR corresponding to the vertical distance between the two Rate-Distortion curves. The other way is to determine the increase/decrease of rate corresponding to the horizontal distance between the two Rate-Distortion curves.

Now there is one slight problem, the two Rate-Distortion curves are not necessarily sampled in the same points (the encoders produces compressed files with different rates). This means that the vertical and horizontal distance can not be obtained from the available rate- and PSNR-values. One
solution to this problem is to interpolate between the sampling-points using splines. By resampling with high resolution (N large) the vertical/horizontal distance can be made arbitrary accurate. The following sections describes how to calculate the drop in PSNR and the increase of rate between two encoders.

Calculating Drop in PSNR

The PSNR drop for an encoder compared to some reference encoder is defined as the average vertical distance between the Rate-Distortion curves. This can be calculated as the area between the curves divided by the corresponding interval of rate. The area, calculated by the integral between the curves, is approximated with a Riemann sum according to figure B.1. The drop in PSNR can thus be calculated as:

$$D = \frac{\sum_{i=1}^{N-1} (PSNR_{ref}(i) - PSNR(i))(rate(i+1) - rate(i))}{rate(N) - rate(1)}$$ (B.1)

Calculating Increase in Rate

The increase in rate for an encoder compared to some reference encoder is defined as the average horizontal distance between the Rate-Distortion curves. Using analogous arguments as above, see figure B.2, the increase of rate is calculated as:

$$I = \frac{\sum_{i=1}^{N-1} (rate_{ref}(i) - rate(i))(PSNR(i+1) - PSNR(i))}{PSNR(N) - PSNR(1)}$$ (B.2)

Often the increase in rate is expressed in per cent of the total rate of the reference. This can be expresses as:

$$R_{ref} = \frac{\sum_{i=1}^{N-1} (rate_{ref}(i))(PSNR(i+1) - PSNR(i))}{PSNR(N) - PSNR(1)}$$ (B.3)

Which finally yields:
Figure B.1: **PSNR drop.** Illustration of how the PSNR drop is calculated.

\[
I\% = 100 \frac{I}{R_{ref}} = 100 \frac{\sum_{i=1}^{N-1} (rate_{ref}(i) - rate(i))(PSNR(i) + 1) - PSNR(i))}{\sum_{i=1}^{N-1} (rate_{ref}(i))(PSNR(i) + 1) - PSNR(i))}
\]  
(B.4)
Figure B.2: **Rate increase.** Illustration of how the rate increase is calculated.
Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/