CHALMERS

Subjectively Optimized HDTV Video Coding

Master of Science Thesis

CHENG XIAOYIN

Department of Signals and Systems
Division of Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2009
Report No. EX025/2009

Subjectively Optimized HDTV Video Coding

Cheng Xiaoyin
caitlin_cheng@hotmail.com

Chalmers University of Technology
Ericsson Research

Table of Content

Table of Abbreviations

AVC	Advanced Video Coding
CODEC	an enCOder/DECoder pair
CABAC	Context-Adaptive Binary Arithmetic Coding
DCT	Discrete Cosine Transform
DQ	Dynamic Quantization
DVD	Digital Versatile Disk
HVS	Human Visual System
KLT	Karhunen-Loeve Transform
MB	Macroblock
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
QP	Quantization Parameter
RDO	Rate Distortion Optimization
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
UVLC	Universal Variable Length Coding
VCEG	Video Coding Experts Group
VLC	Variable Length Coding

Abstract

Video compression deals with compact representations of video signals for storage and transmission. It takes advantage of features of the human visual system (HVS) for a more efficient compression.

When people are looking at a picture or watching a video sequence, human perception is more sensitive to distortion on homogeneous areas than on textured parts. Therefore, adaptive quantization methods are carried out in video coding to improve the compression. A lower quantization parameter (QP) is assigned to macroblocks having smooth content to get a high fidelity compression, while a higher QP is used for the rest.

There are some inefficiencies with the application of adaptive QP. Macroblocks having a half smooth and half textured content are generally assigned with a higher QP. This results in more obvious and heavier distortion on the homogeneous half. These macroblocks are often found around figures against a smooth background. Hence distortion frequently appears around figures in the form of ringing artifacts.

In this thesis, pixels covering the smooth part of partially-textured macroblocks are denoted as “target pixels”. The main objective in this thesis is to reduce ringing artifacts by compensating target pixels for distortion. It is achieved by devising a pixel classification algorithm and a distortion compensation algorithm- The former is used to select target pixels, and the latter is applied to suppress ringing artifacts. The thesis algorithm is implemented in the Rate-Distortion (RDO) mode decision part in an H.264 encoder. By using this algorithm, RDO intends to select higher bit-cost modes for partially textured macroblocks (which contain target pixels), such that the distortion for target pixels is reduced. This results in less distortion and a reduced amount of ringing artifacts.

Subjective assessments show that, for some characteristic sequences (containing textured figures against smooth areas), ringing artifacts around stable figures such as logos or scoreboards are efficiently reduced. However, for highly moving irregular figures, the improvement is comparatively small.

Keywords: H.264, Ringing artifacts, RDO, Pixel classification, Subjective assessment

Acknowledgements

Above all, I want to show my respects and deepest appreciation to my supervisor, Rickard Sjöberg, for giving me the opportunity to work in the multimedia research group, for teaching me much more than I can learn from the class, and for continuously encouraging me with his patient guidance and friendly communications, throughout the whole period of time.

I'm also grateful to Per Fröjdh, the manager of the visual group, and all the members in the visual group, for everyone's kindly help and contribution to such nice working environment. Thanks to Fred, Su Hui and Wang Jia, for helping me solve programming and technical problems.

I would also like to thank to my examiner Prof. Mikael Persson and Markus Johansson at Chalmers, for supporting me to be engaged in the thesis work that I'm dreaming of, though it is not so related to biomedicine.

So many thanks to my landlords but also my best friends, Victor and Nan. They show me how to live a meaningful and healthy life. Their encouragements make me brave to face frustrations and failures.

Last but not least, I want to thanks all my friends and families for their trust, constant support, and tolerance.

Cheng Xiaoyin
March 29th, 2009

Outline

Chapter 1: The theoretical background is presented. An introduction to video compression and relevant basic concepts are given, accompanied with a thorough description of rate distortion optimization. The definition and measurements of video quality are also discussed.

Chapter 2: The motivation for this thesis is explained. Problems are specified and a brief discussion about why the problem cannot be solved by currently methods is given.

Chapter 3: The classification algorithm is described in detail.

Chapter 4: A feasibility analysis is presented prior to the introduction of the distortion compensation algorithm. At the end of this chapter, the general methodology combining both the classification algorithm and the distortion compensation algorithm is summarized.

Chapter 5: The fine-tuning of algorithm parameters is explained. Then actual simulations are described and results are subjectively evaluated.

Chapter 6: The conclusions of the whole thesis work are made, followed by a proposal of possible future work.

Chapter 1 Introduction

In this chapter, the theoretical background needed to understand the rest of this thesis is presented. Throughout this chapter, [1], [2] and [3] has been used as reference literatures.

1.1 Video Compression

Uncompressed video sequences contain a huge amount of information that is overwhelming for most transmission or storage environments. For instance, nowadays internet throughput rates cannot handle uncompressed video in real time; A Digital Versatile Disk (DVD) is only able to store one minute of uncompressed high-definition video. To solve this problem, video sequences are compressed, enabling more convenient storage and efficient transmission.

Video compression involves a compliant pair of system --- an encoder and a decoder. The encoder works as a compressor that condenses raw data into a smaller number of bits by removing redundancy form the source. Contrarily, the decoder converts compressed information to a representation of the original. The combination system of an encoder/decoder pair is noted as a video CODEC (enCOder/DECoder) system.

1.2 H.264 Standard

Video compression techniques have been highly developed during decades, accompanied with a number of key international standards, such as the MPEG and H.26x series.

H.264 is the latest standard, it is also known as MPEG-4 Part 10, or MPEG-4 AVC (Advanced Video Coding). It was published in early 2003 by the ITU-T Video Coding Experts Group (VCEG), together with the Moving Picture Experts Group (MPEG). [1]

H.264 intends to provide significantly better compression than any previous standard. It also aims at giving high flexibility so that the standard is able to be applied to a wide variety of applications on a wide variety of systems, such as low and high resolution, low and high bit rate video, DVD storage, RTP/IP packet networks, and ITU-T multimedia telephony systems [4].

Basically, two things are defined in H.264: a coded syntax describing compressed visual data and the method for decoding the syntax and reconstructing the visual information. The standard guarantees that pairs of encoders and decoders are bit-exact. Note that the standard does not define the encoding process but only the syntax of an encoder video bit stream and the decoding process.

1.3 Colour Spaces

Most digital video applications rely on the display of colour video. This requires a mechanism to capture and represent colour information. The mechanism that defines colour and brightness (luminance or luma) is depicted as a colour space. Generally, a colour space consists of three colour planes.

1.3.1 RGB

In the RGB colour space, an image is made up by Red, Green and Blue colour planes. Almost all colours can be created by combining red, green and blue in varying proportions [3]. Though RGB is the most commonly used colour space, it does not correspond well to human perception of colours [3].

1.3.2 YCbCr

YCbCr is a popular colour space where the Y layer represents the brightness component. The Cb, and Cr layers represent colour information. They can be calculated from R, G and B [1]:

$$\begin{aligned} Y &= k_r R + (1 - k_b - k_r) G + k_b B \\ Cb &= \frac{0.5}{1 - k_b} (B - Y) \\ Cr &= \frac{0.5}{1 - k_r} (R - Y) \end{aligned} \qquad (1.1)$$

where k are weighting factors. ITU-R recommendation BT.601 [1] defines k_b = 0.114 and k_r = 0.299.

R,G and B components can be calculated from Y, Cb and Cr:

$$\begin{aligned} R &= Y + \frac{1 - k_r}{0.5} Cr \\ G &= Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} Cb - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} Cr \\ B &= Y + \frac{1 - k_b}{0.5} Cb \end{aligned} \qquad (1.2)$$

1.3.3 4:2:0 Sampling Format

YCbCr is used instead of RGB in video coding because the former is more compression efficient. The Human visual system (HVS) is more sensitive to luminance than colour, so Cr and Cb components can be represented with a lower resolution than Y, resulting in a reduced bit cost.

A popular sampling format in H.264 is 4:2:0, which means that the chroma components (Cb and Cr) each have half the horizontal and vertical resolution of Y. For example, if there are MxN samples in Y layer, there are each 0.25xMxN Cb and Cr samples. (4 luminance samples share a pair of Cb and Cr samples)

1.4 Video Coding Concepts

In this section, a general introduction of the video CODEC system is given, followed by a detailed discussion of a few particular functional units (e.g. motion estimation and compensation, transformation, quantization).

1.4.1 Redundancy

The source video is compressed by removing redundancy, which mainly has three types: statistical redundancy, temporal redundancy and spatial redundancy.

In lossless compression systems, statistical redundancy is removed and the reconstructed signal is identical to the original one without any loss. However, this method can only achieve a modest amount of compression of video signals.

In practice, a hybrid CODEC system based on lossy compression is commonly used. Besides removing statistical redundancy, it also reduces redundancy in temporal and spatial domains, taking advantages of limitations of the human visual system (HVS). For example, consider a video sequence captured by a digital camera at high frame rate. Neighbouring frames may have little differences and smooth content of a frame has small variations in pixel values. Differences or variations that are unnoticeable by human eyes make up what is called "redundancy".

1.4.2 Encoder

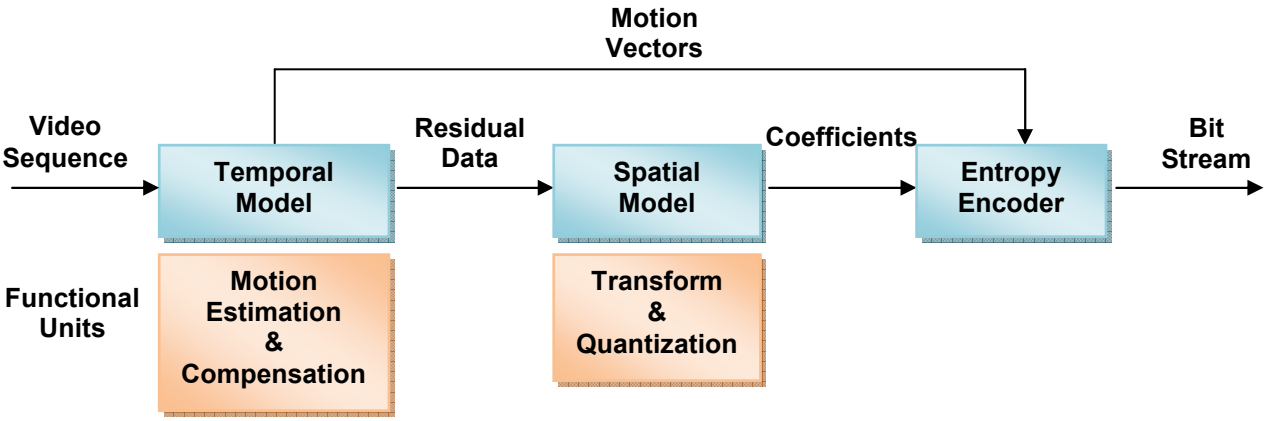


Figure 1.1 *Video encoder*

A video encoder (video encoder) includes three basic models: a temporal model, a spatial model and an entropy encoder. Each of them contains key functional units (Only those mentioned in this thesis are shown in Figure 1.1).

- **Temporal Model**

Uncompressed video is fed to the temporal model, whose function is to search for similarities between adjacent video frames to reduce temporal redundancy. A prediction of the current frame is generated from previous or future frames (or from the combination of both previous and future frames). The prediction is often improved by compensating motion differences between relevant frames (by motion estimation and compensation). In the end, by subtracting the prediction form the current frame, a residual frame is generated and transferred to the spatial model, together with a set of motion vectors describing the motion.

- **Spatial Model**

The spatial model is applied to find similarities between neighbouring pixels within a residual frame, reducing spatial redundancy. This is achieved by transforming the residual frame into another domain, in which the residual data is represented by coefficients that are more

independent with each other. Then insignificant coefficients values are removed through quantization, leaving only a relevant small number of significant coefficients to be sent to the entropy encoder.

- **Entropy Encoder**

Quantized coefficients and motion vectors are compressed by the entropy encoder. It removes statistical redundancy by representing commonly-occurring vectors and coefficients by short binary codes (e.g. CABAC, UVLC, VLC). Finally a compressed bit stream is produced, whose format is standardized by video compression standards such as H.264. That includes header information, coded motion vector parameters and coded residual coefficients.

1.4.3 Decoder

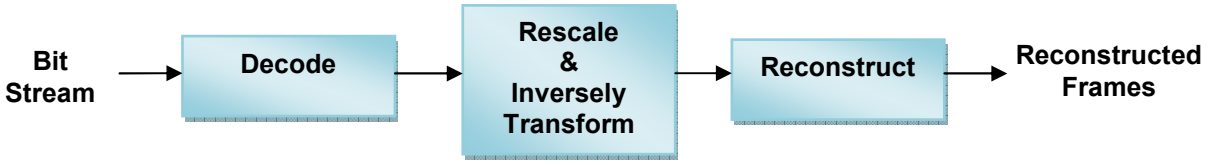


Figure 1.2 *Video decoder*

A decoder (Figure 1.2) works similar to the encoder, but in a reverse way. When a standard bit stream comes into the decoder, the motion vectors and quantized transform coefficients are firstly decoded. Then the coefficients are rescaled (similar but not the same as the original ones since quantization is a lossy process) and inversely transformed back to the original domain, restoring residual data. Eventually, residual data will be added to a previously reconstructed reference frame, according to the decoded motion vectors, to reconstruct the current frame. The currently reconstructed frame is stored and might be used as reference frame for decoding future frames.

1.4.4 Block-based Video Coding

In a block–base video CODEC, a video frame is encoded block by block. The currently encoded frame is partitioned into MxN rectangular sections or blocks. A block becomes the basic unit for coding. Generally, a macroblock, corresponding to a 16x16 pixel region of a frame, is used in many important video coding standards such as MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263 and H.264 [1, 2].

1.4.5 Block-Based Motion Estimation and Compensation

Block-based motion estimation and compensation plays a crucial role for reducing the amount of residual information [1,3]. The motion estimation is carried out to find the best matching sample region in the reference frame (Figure1.3).

A popular matching criterion defines the best match to be the reference region that minimizes the energy in the residual.

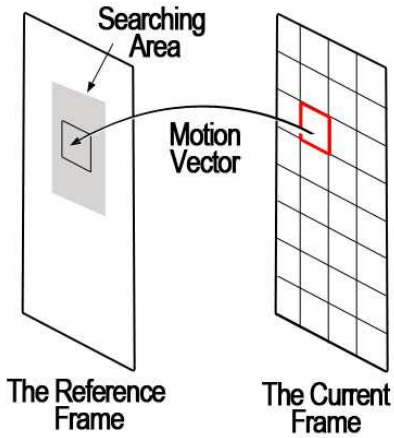


Figure 1.3 *Block-based motion estimation and compensation*

The offset between current macroblock and the position of the reference region is denoted as motion vector of the macroblock. The set of motion vectors will be coded in entropy encoder, as well as the quantized coefficients of the macroblock residual.

1.4.6 Prediction Modes in H.264

In H.264, three basic prediction modes are defined: the skip mode, inter-prediction modes and intra-prediction modes.

- **Skip mode**

A macroblock can be skipped if there are no or only little changes from the corresponding reference region. The prediction of the current macroblock is copied from the reference frame. The macroblock is coded with one bit only that indicates the skip mode. No motion vectors or transformed coefficients are transmitted.

- **Inter-prediction modes**

Here motion vectors are signalled that describes how the prediction is formed from previously coded pictures. The macroblocks might be partitioned. A good match of each partition is found and all partitions together make up the prediction. Each partition has its own motion vectors.

Available partitions (Figure 1.4) of a macroblock in H.264 are: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4.

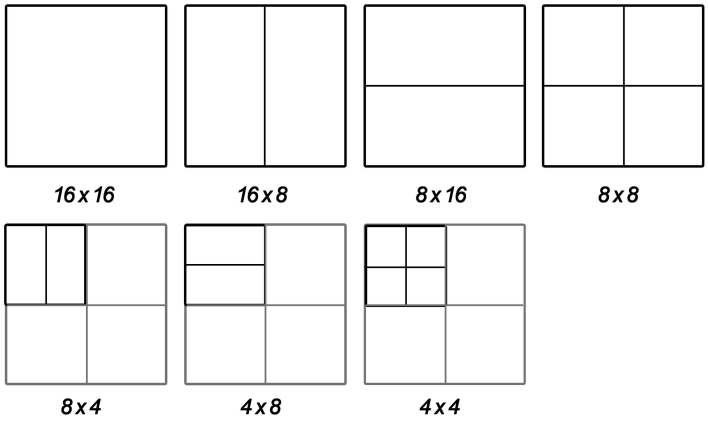


Figure 1.4 *Possible Block Splits*

- **Intra-prediction modes**

Intra is the optimal alternative when no good motion match is found. Instead of referring to reference frames, the current macroblock is predicted from surrounding pixels of the same frame. As depicted in Figure 1.5, the current macroblock (marked by a red rectangle) is predicted from 17 surrounding pixels (marked by Roman numerals.)

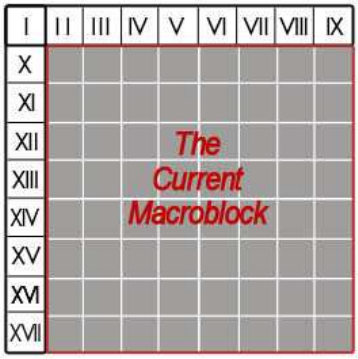


Figure 1.5 *Intra-prediction*

If intra-prediction is chosen, the macroblock is split into 16x16 or 4x4. There are 4 intra-prediction modes for 16x16 blocks (macroblocks) and 9 for 4x4 ones [1].

1.4.7 Transform

The purpose of the transform is to convert the residual into a transform domain, where the transformed data is more de-correlated (has minimal inter-dependence) and compact (most of the energy in the transformed data is concentrated into a small number of values).

Many transforms has been proposed for video coding. Examples of block-based transforms include the Karhunen-Loeve Transform (KLT), Singular Value Decomposition (SVD) and the ever-popular Discrete Cosine Transform (DCT) [1].

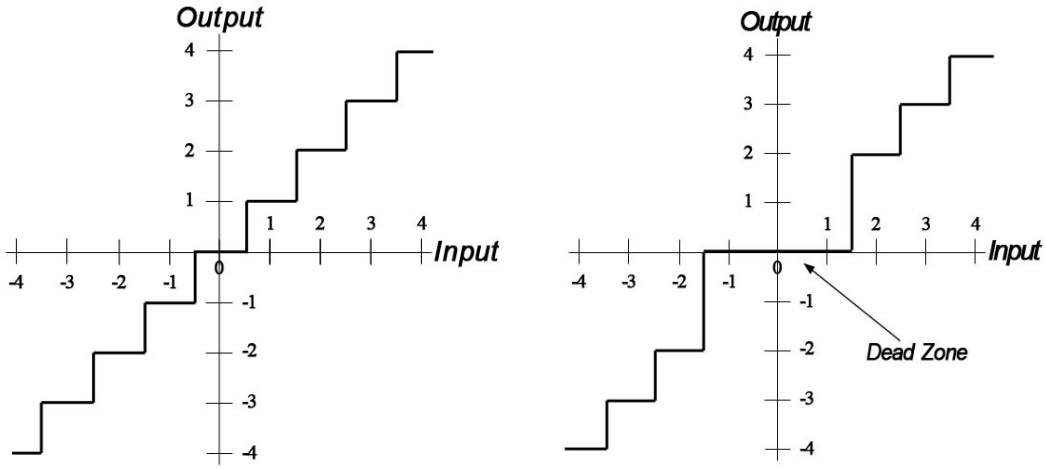
1.4.8 Quantization

Quantization is a lossy compression process. A quantiser maps a digital signal with a set of values X to a quantized signal with a reduced number of values Y. Insignificant values are set to zero, and a reduced range of possible values is obtained, resulting in lower bit cost for coding the quantized coefficients [1].

A general example of a quantiser is:

$$FQ = round(\frac{X}{QP}) \; ; \; Y = FQ \cdot QP \qquad (1.3)$$

where QP (Quantization Parameter) refers to quantization “step size”. The quantized output levels are spaced at uniform intervals of QP. As shown in Figure 1.6, there are two examples of scalar quantizers. (a) is a linear quantiser and (b) is a non-linear one which has a ‘dead zone’ around zero. Small values in the dead zone are mapped to zero.



(a) linear *(b) non-linear*

Figure 1.6 *Linear (a) and Non-linear (b) scalar quantisers*

QP	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Q_{step}	0.625	0.6875	0.8125	0.875	1	1.125	1.25
QP	**7**	**8**	**9**	**10**	**11**	**12**	**…**
Q_{step}	1.375	1.625	1.75	2	2.25	2.5	…
QP	**18**	**…**	**24**	**…**	**30**	**…**	**36**
Q_{step}	5	…	10	…	20	…	40
QP	**…**	**42**	**…**	**48**	**…**	**51**	
Q_{step}	…	80	…	160	…	224	

Table 1.1 *Quantization Parameters and Quantiser step size*

H.264 supports a total of 52 values of quantiser step size (Qstep), which are indexed by the QP. The value of step size doubles for every increment of 6 in QP. For instance, as seen in the Table1.1, when quantiser step increases from 1 to 2, the index QP changes from 4 to 10, and ∆QP (DQ) equals to 6.

The QP used for each picture is called “base QP” or “picture QP. If the QP changes for macroblocks in a picture, the base QP is the average QP for the picture. Using “fixed QP” means that all macroblocks in a sequence are encoded with a fixed constant QP. In contrast, a “rate control” method may be used to control the bit rate by adjusting the base QP between pictures. When the bit rate is exceeding the target bit rate, the base QP is increased by the rate control. In this case, several base QPs are used when encoding one sequence.

Quantization is an essential functional unit for the CODEC system to adjust the balance between compressed video fidelity with bit cost. Encoding with a higher QP results in less bit cost but poorer quality and vice versa.

1.4.9 Adaptive Quantization Parameter

The human visual system (HVS) is more sensitive to distortion in a homogeneous area, while the same amount of distortion in a highly textured area will be unnoticed. Adaptive QP

methods used in video CODEC systems utilize this which, results in better subjective video compression.

The adaptive QP method first categorizes the macroblocks in a frame. Then for each category, a delta QP (DQ) is assigned, which can be either a positive or negative integer. The QP of a category is obtained by adding the DQ to the base QP of the frame. So macroblocks in the same category have the same macroblock QP, while macroblocks in different categories have different macroblock QP (MB QP).

The core of an adaptive QP algorithm is to automatically select macroblocks that is comparatively homogeneous in content or is less textured. This can be achieved by implementing an activity matrix for each macroblock:

$$Activity = \sum_{x=0}^{14}\sum_{y=0}^{15} | Y_{x,y} - Y_{x+1,y} | + \sum_{x=0}^{15}\sum_{y=0}^{14} | Y_{x,y} - Y_{x,y+1} | \qquad (1.4)$$

Where Y is the pixel array of a macroblock, and (x, y) stands for the pixel coordinates.

Formula 1.4 says that the activity is the sum of neighbouring pixel difference in a macroblock. If the macroblock has a smooth content (i.e. pixels of it are similar), it gets a small activity. In contrast, if the macroblock is textured or partial-textured (adjacent pixels differ from each other), it has a comparatively large activity.

Activities of all macroblocks are sorted from minimum to maximum. A threshold was manually set for categorization. In Figure 1.7, macroblocks having activities under the threshold, are marked in blue and classified into a low category, denoted as “low activity macroblocks”. High category/activity macroblocks are unmarked. The preferred threshold was selected so that low activity macroblocks almost covered all smooth areas of each frame.

To suppress the most obvious distortion effects in less textured macroblocks, lower MB QP is applied by adding a negative ΔQP (DQ) to the base QP to get lower distortion. This, however, generates more bits. (But, since they are less textured, these macroblocks generate less transformed coefficients so that naturally they are easier to compress with a low bit cost.) Contrarily, considering the average bit cost for each frame, highly textured macroblocks are coded with less bits and higher MB QP (the base QP plus a positive DQ), resulting in more distortion. However, it is hard to observe the distortion increase since it is masked by the texture. To summarize, the adaptive QP method is used to spend more bits on less textured macroblocks at the expense of increasing distortion on textured areas.

Figure1.7 *Macroblocks coloured in blue belongs to low category*

1.5 Rate-Distortion Optimization

Classical rate distortion theory is described in Shannon's seminal work and discussed in information and communication theory [5]. It has implications for the video compression task: how to find a rate-distortion trade-off, that is, a compromise between bit cost and compression fidelity.

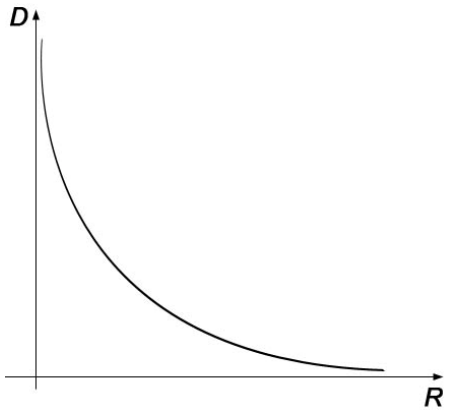


Figure 1.8 *The Rate-Distortion Curve*

The rate distortion theory points out that rate D(R) is a convex function of the bit cost R, and this function is successively monotonously decreasing (Figure 1.8) [7].

1.5.1 Distortion Measurements

Distortion is a metric of the difference between the source signal and decoded signal. In practice, it is often objectively described by one of the following functions [7, 8]:

- Sum of Absolute Difference:

$$SAD_{\mathrm{A}}(R,S)=\sum_{i\in \mathrm{A}}\left|R(i)-S(i)\right| \qquad (1.5)$$

- Sum of Squared Differences:

$$SSD_{\mathrm{A}}(R,S)=\sum_{i\in \mathrm{A}}\left|R(i)-S(i)\right|^{2} \qquad (1.6)$$

- Mean Squared Error:

$$MSE_{\mathrm{A}}(R,S)=\frac{1}{\left|\mathrm{A}\right|}SSD_{\mathrm{A}}(R,S) \qquad (1.7)$$

Where “A” is a set of pixels in the relevant area and “|A|” is the cardinality of the set “A” (number of elements of the set A), “i” represents the pixel number, and “R” and “S” stands for arrays of pixel values of the reconstructed area and the source respectively.

1.5.2 Operational Rate-Distortion Boundary

R-D performance is the fundamental trade-off in the design of any lossy compression system [6]. However, it is almost impossible to find the ideal R-D curve (as shown in Figure 1.8, due to calculation complexity and other problems [7].

For practical video coding, an alternative method is to define an operational rate-distortion bound, which is obtained by plotting R-D operating points: Before actual encoding, every possible combination of coding parameters (QP, prediction mode, and so on) is tried to pseudo-code the current macroblock. The distortion (D) and bit cost (R) are recorded. Together they specify an operating point in the rate-distortion chart. Various coding options result in a great number of R-D points, as demonstrated in Figure 1.9.

This operational R-D performance boundary is then defined by the convex hull of the set of operating points (the black dashed curve in Figure1.9). It allows us to distinguish between the best achievable operating points (nearer to or on the curve) and those that are suboptimal.

It can be noted that modes providing higher quality (such as intra-predictions, inter-predictions with more splits) cost more bits and less distortion. The operating points of these modes are generally located close to the R axis and away from the D axis. Vice versa, inter-prediction modes with less splits, e.g. P16x16, result in lower video compression fidelity and lower bit cost. These operating points are close to axis D and far from the R axis.

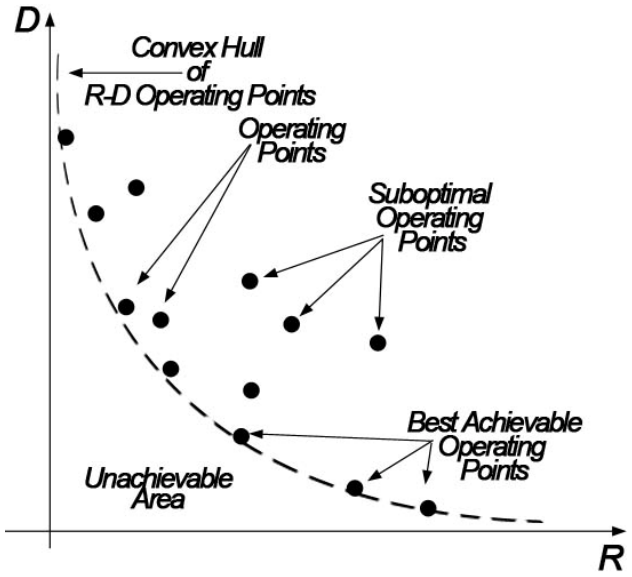


Figure 1.9 *Operational Rate-Distortion Curve of a Given Source*

1.5.3 Rate-Distortion Optimization

It should be pointed out that with different sources (e.g. different video sequences, different scene content), the operational rate-distortion curve varies. Even with the same source coding at different bit rates, different coding options show different efficiency.

In reality, physical channels have different limitations on the bit rate. Under a certain bit rate, choosing the best coding option in rate-distortion sense can be read as: subject to a constrained R_C on the bit cost R, minimize distortion D:

$$\min\{D\},\ \ subject\ to\ R < R_c \qquad (1.8)$$

The optimization task can be elegantly solved by a popular method known as Lagrangian optimization [8], where a distortion term is weighted against a rate term, and a Lagrange multiplier λ is introduced:

$$\min\{J\},\ \ where\ J = D + \lambda R,\ \ \lambda \geq 0 \qquad (1.9)$$

For each coding unit (i.e. a macroblock or a frame), with a particular value of the Lagrange multiplier λ, the Lagrangian cost J is minimized by the optimal coding option (represented by best operating points in R-D coordinate). Figure 1.10 gives a graphical explanation of the Lagrangian cost formula. The chosen operating point is the one “hit” by a “plane wave” of slope λ, since it has the smallest value of the Lagrange cost J.

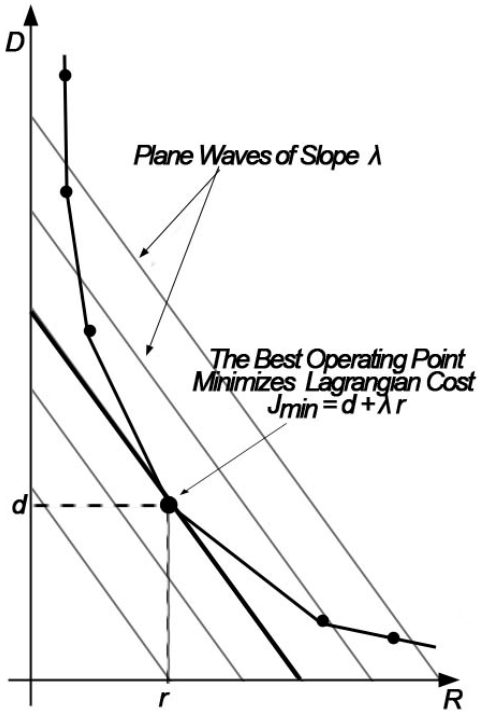


Figure 1.10 *Search for the best operating point;*

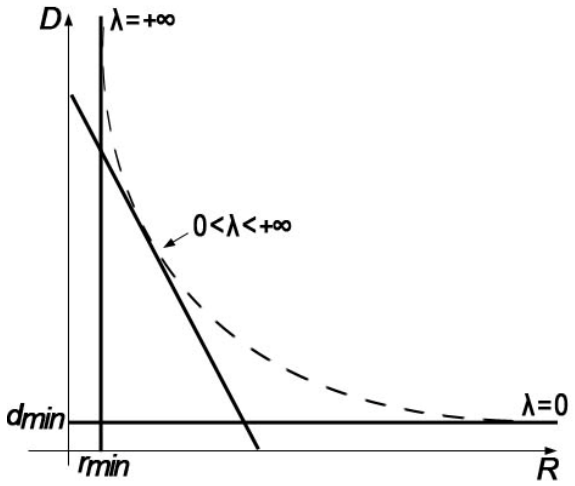


Figure 1.11 *Using Lagrange multiplier λ to choose R-D trade-off point*

The Lagrange multiplier λ will select a specific trade-off point between rate and distortion. As demonstrated in Figure 1.11, when λ=0, minimizing the Lagrangian cost is equivalent to minimizing the distortion. Conversely, minimizing the Lagrangian cost when λ = +∞, is equivalent to minimizing the rate.

1.5.4 RDO for Mode Decision

Rate distortion optimization (RDO) can be applied in the CODEC for optimizing motion estimation, reference frame selection and mode decision [6]. Employing RDO for selection of modes is here discussed in detail:

As mentioned in the previous section 1.4.6, the available modes in H.264 are:

- Skip mode;
- Inter-prediction modes (P-prediction):
  Block size selection: { P16x16, P16x8, P8x16, P8x8, P8x4, P4x8, P4x4 };
- Intra-prediction modes (I-Prediction): { I16x16, I4x4 }.

The basic principle of mode decision in H.264 is presented here: First, the current macroblock is pseudo-coded. Then the Lagrangian costs J = D + λR is calculated for each possible mode. The mode that gives the minimum J is then selected for final encoding. As shown in Figure 1.12, the distortion D is measured by calculating the SSD between the reconstructed and original macroblock pixels. R refers to the number of bits spent for coding the entire macroblock (including transformed residual coefficients and motion vectors).

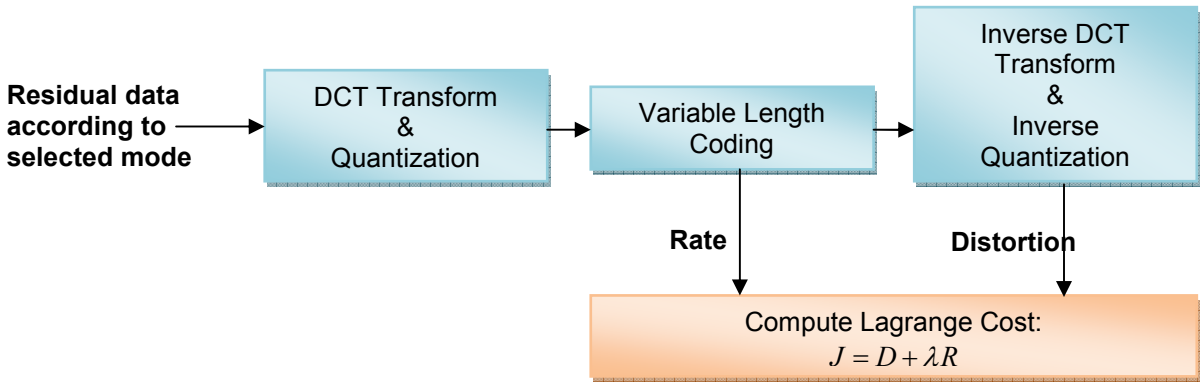


Figure 1.12 *The process of calculating Lagrange cost in H.264*

The Lagrange multiplier λ is based on the QP value. Experiment results show that using only one λ for all QPs result in low compression efficiency. Practical λ=f(QP) functions are commonly monotonically increasing functions as shown in Figure 1.13.

An example of λ=f(QP) used for H.264 coding specified in [9] is :

$$\lambda = 0.85 \times 2^{(QP-12)/3}, \quad where \ 0 \le QP \le 51 \qquad (1.10)$$

All experiments presented in this thesis are using this function for computing λ from QP. (Numerical values of λ are included in Appendix A).

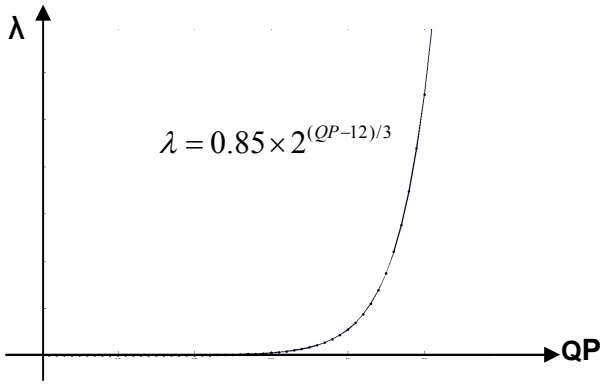


Figure 1.13 *QP versus Lambda (*$\lambda = 0.85 \times 2^{(QP-12)/3}$*)*

When high QP is assigned, which results in a large λ, the Lagrangian cost J = D + λR weights more on bit rate, which means R becomes the dominating factor. Modes that cost less bits then have higher probability to be selected. Vice versa, a smaller QP and a smaller λ make D more important. Modes (such as intra prediction) resulting in a low distortion may then be chosen.

1.6 Video CODEC Performance

In this section, subjective and objective approaches to measure compressed video quality are presented, accompanied with methods to evaluate performance of video communication system.

1.6.1 Subjectively Quality Measurement

Subjective quality may be influenced by many factors: spatial and temporal fidelity of the test video sequence, the viewing environment, the observer's state of mind and the extent to which the viewer interacts with the visual scene [1,3]. These factors make it very difficult to measure visual quality precisely and quantitavely.

1.6.2 Objective Quality Measurement: PSNR

In contrast to subjective quality measurement, developers in the field of video compression rely heavily on objective quality measurements. The most well known measure is Peak Signal to Noise Ratio (PSNR):

$$PSNR_{dB} = 10\log_{10}\frac{(255)^2}{MSE} \qquad (1.11)$$

It depends on MSE (see 1.4.1). Easy and quick calculation makes PSNR a very popular measure.

1.6.3 Video CODEC Performance Evaluation

The video CODEC performance is a trade-off between three factors: quality, compressed bit rate and computational cost [1]. If no requirement exists on encoding in real time, the computational cost is of less importance.

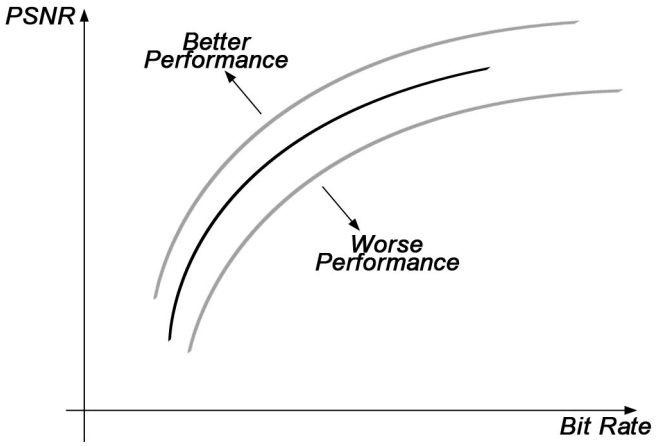


Figure 1.14 *The rate-distortion performance curve*

A characteristic performance curve (Figure 1.14) can be obtained by plotting PSNR against coded bit rate. The objective video quality (PSNR) drops at an increasing rate with a decreasing bit rate.

Chapter 2 Problem Specification

In this chapter, distortion in the form of ringing is described. This type of distortion can be clearly observed in the NBC_Clip6 sequence, which is the main test sequence in this thesis. NBC_Clip6 is a short 1080i video clip of an American football game. It is made up by 288 frames and lasts for 11.52 seconds at 25 Hz field rate. The sequence starts with a camera pan that turns into a zoom. Football players run at a high speed on the grass field. An overlay scoreboard is shown at the bottom of the frames throughout the whole sequence.

2.1 Ringing artifacts and Target Pixels

(a) ***(b)*** ***(c)***

Figure 2.1 *Ringing artifacts caused by block effects*
(a) Ringing artifacts around players;(b)Ringing artifacts around bottom logo and scoreboard;
(c)Ringing artifacts around shapes (lines and circles)

In Figure 2.1, macroblocks are outlined by black lines. Blue macroblocks are low activity macroblocks as categorized by the adaptive QP method. Macroblocks in original colour are in the high activity category.

Low activity macroblocks are assigned a lower QP. They are represented with comparatively high fidelity at the expense of increased distortions on high activity macroblocks (see section 1.4.9). High activity macroblocks are either fully textured or partially textured. For the partial-textured macroblocks, the distortion is unnoticeable in the textured part while the distortion is obvious in the homogenous part, often seen as ringing artifacts. In Figure 2.1, three high activity macroblocks are pointed out by red rectangles. Only a small portion of these macroblocks are textured. Distortion can be clearly seen in the smooth part of these macroblocks.

Ringing artifacts are mostly found around high textured figures against homogeneous areas. For instance, in the NBC_Clip6 video it can be observed around players, above the scoreboard and around shapes on the smooth grass field.

2.2 Select target pixels

A pixel is denoted as a "**target pixel**", only if it is in a partially-textured macroblock and the pixel belongs to the homogeneous part of the macroblock. To reduce ringing artifacts, only the target pixels need improvement. No extra bits should be spent on improving the textured pixels. No matter how the distortion is suppressed for target pixels, the first objective is to pick them out.

The category of a macroblock is based on macroblock activity as explained in section 1.4.9. Macroblocks having small activities are classified into a low category. To include target pixels into the low category, the activity matrix should be calculated on the pixel level. The definition of pixel activity, together with a detailed discussion on the pixel classification algorithm, is described in chapter 3.

2.3 Breaking down QP into pixel level

To improve the quality of target pixels, an algorithm working on the pixel level is required. Unfortunately, it is impossible to break down QP and assign a lower QP for target pixels and a higher QP for the rest of the pixels. Currently in H.264, QP can only be set on the macroblock level.

In chapter 4, an experiment is first presented. The results show that changing the Lagrangian multiplier λ gives a similar effect to changing QP and that changing the distortion of the Lagrangian cost function gives the similar effect as λ changes. Based on these results, a distortion compensation algorithm is proposed.

Chapter 3 Pixel Classification Algorithm

In this chapter, a pixel classification algorithm is proposed. A pixel activity is defined and used to classify pixels into two categories: low activity pixels and high activity pixels. To pick out target pixels (which are low activity pixels in high activity macroblocks), low activity pixels belonging to low category macroblocks are excluded.

Figure 3.1 shows the relationship between macroblock categories and pixel categories

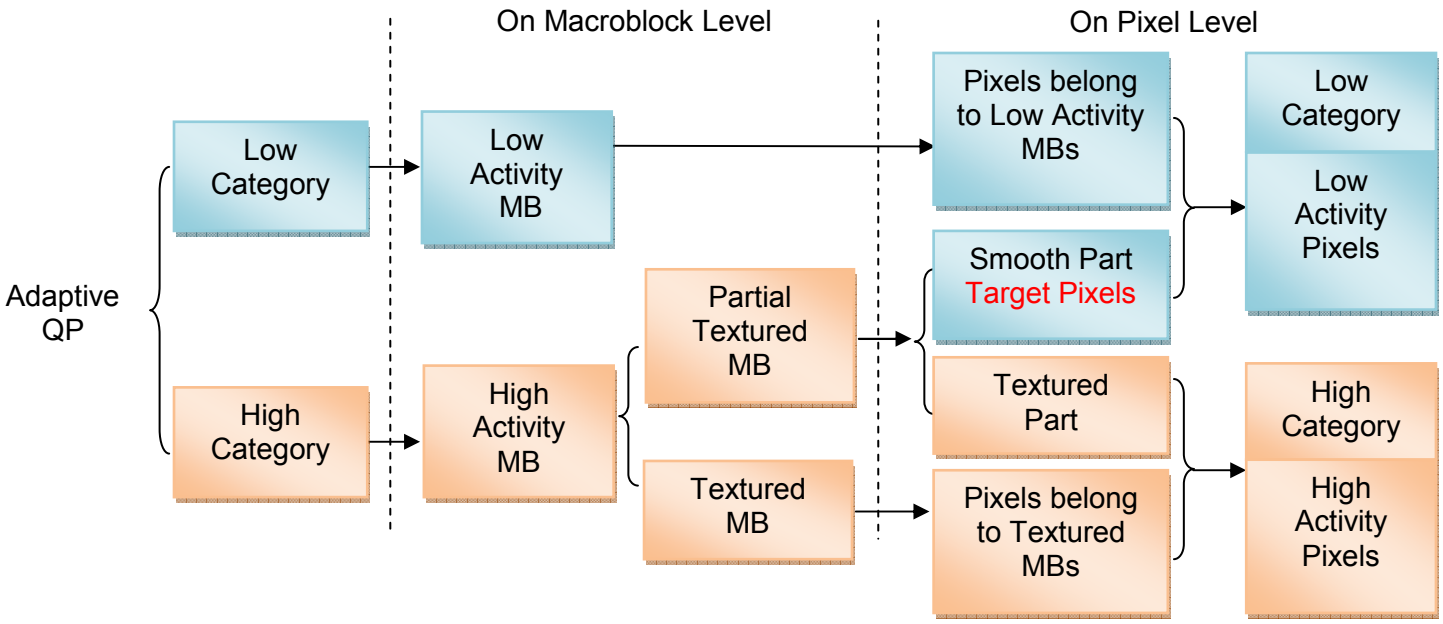


Figure 3.1 *Low/High activity MBs and Low/High activity pixels*

3.1 Overlapped Block Activity

In 1.4.9 an activity matrix was introduced:

$$Activity = \sum_{x=0}^{14}\sum_{y=0}^{15} | Y_{x,y} - Y_{x+1,y} | + \sum_{x=0}^{15}\sum_{y=0}^{14} | Y_{x,y} - Y_{x,y+1} | \qquad (1.4)$$

In this thesis, an activity matrix is applied to a series of overlapped blocks instead of macroblocks. As shown in Figure 3.2, each block represents a pixel, and 256 blocks make up an overlapped block. Each pixel is the origo of one particular overlapped block. For instance, pixel no.7 is the origo of the overlapped block shown in pink. The activity of an overlapped macroblock is calculated by formula 1.4, using pixels which make up this overlapped block.

The size of the overlapped blocks can be varied (16x16, 8x8 or 4x4) for different scene content. In this thesis, it is manually set before encoding. For an MxN image, there are (M-bs+1)x(N-bs+1) overlapped blocks, where “bs” denotes the overlapped block size.

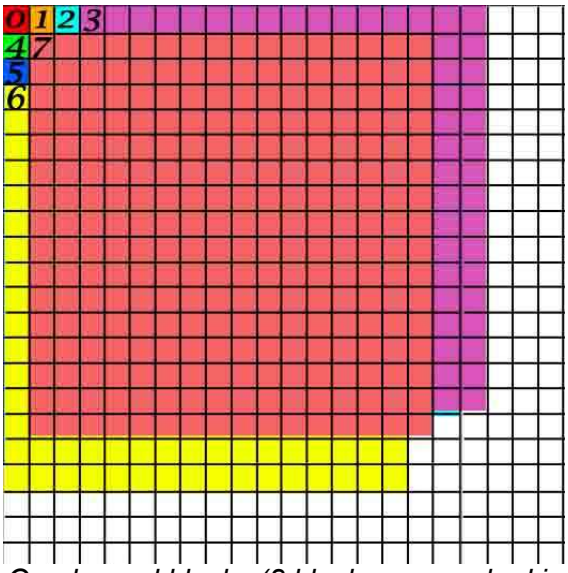


Figure 3.2 *Overlapped blocks (8 blocks are marked in this figure)*

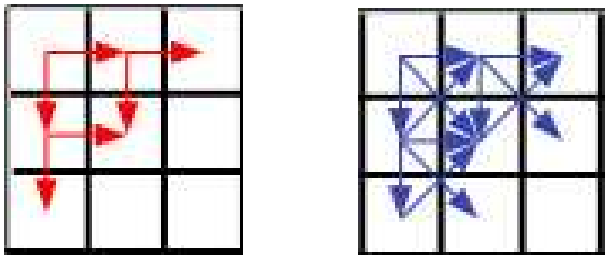
3.1.1 Improvements made to the activity matrix

The activity matrix used in this thesis is modified to fit multiple block size and improved by adding two diagonal activities:

$$Activity = \sum_{x=0}^{bs-2}\sum_{y=0}^{bs-1}|Y_{x,y}-Y_{x+1,y}|+\sum_{x=0}^{bs-1}\sum_{y=0}^{bs-2}|Y_{x,y}-Y_{x,y+1}|+\sum_{x=0}^{bs-1}\sum_{y=0}^{bs-1}\left|Y_{x,y}-Y_{x+1,y+1}\right|+\sum_{x=bs-1}^{0}\sum_{y=0}^{bs-1}\left|Y_{x,y}-Y_{x-1,y+1}\right| \quad (3.1)$$

Figure 3.3 is a graphical explanation of formula (1.4) and (3.1). The activity acts as a detector of pixel differences. In (1.4), the detector is only sensitive to vertical or horizontal differences between neighbouring pixels. With formula (3.1), differences from four directions are considered, improving precision and sensitivity of the detector.

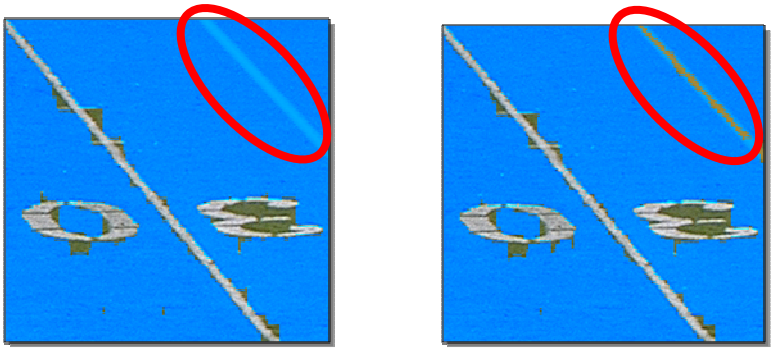
.



(a) ***(b)***

Figure3.3 *(a) Original method to calculate activity; (b) Two diagonal activities have been added; Here, each block represents a pixel.*

3.1.2 Comparative experiments:



(a) ***(b)***

Figure3.4 *pixel classification process with*
(a) Original activity algorithm; (b) modified activity algorithm;
Pixels in low category are marked in blue. Red circles point out the main difference.

Figure 3.4 shows two slices cut from clips that has been processed with the original and modified activity functions respectively. With exception of the selected activity algorithm, all conditions are the same. Comparing (a) and (b), the original activity detector misses the half-transparent yellow line (included in red circles), while the modified one does not.

3.2 Pixel Activity and Classification

3.2.1. Definition of Pixel Activity

A pixel may be included in a number of overlapped blocks. Activity of a pixel is defined as the minimum value among activities of all overlapped blocks which contain this pixel. This definition can be written as:

$$PA(x,y) = \min\{OBA\},\ \ (x,y) \in \{OB\} \qquad (3.2)$$

where PA denotes activity of a pixel (x,y); OBA denotes the set of activities of overlapped blocks (OB) which contain the pixel (x,y).

3.2.2. Pixel Classification

The proposed pixel classification process is carried out as follows:

- The activities of all overlapped blocks are calculated.
- For each pixel, its pixel activity is set to the minimum overlapped block activity.
- Pixel activities are sorted from the minimum to the maximum.
- A threshold is set to control the number of pixels in the low category. E.g. a threshold of 70 means that the first 70 percent pixels are classified as low activity pixels.
- Set the “status” of each pixel. The status is set to 1 for low activity pixels that do not belong to a low category macroblock. All other pixels have status 0.

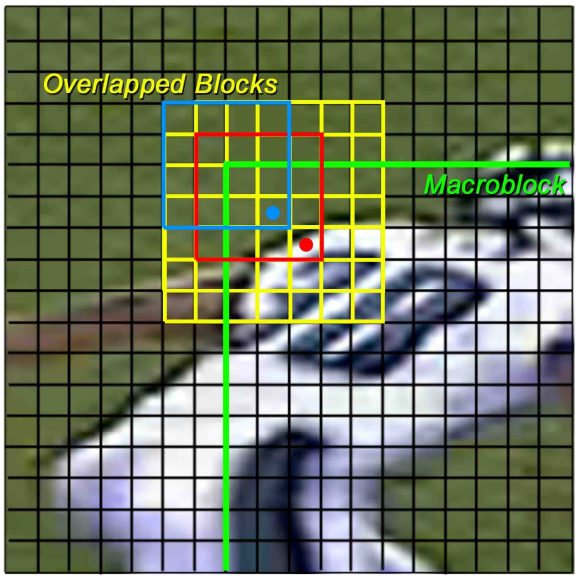


Figure 3.5 *Introduce pixel activity for classification.*
(Overlapped block size of 4x4 is used in this figure.
For the blue pixel, all involved overlapped blocks are contoured in yellow. The pixel activity of the blue pixel is set to the block activity of the blue block, since it has the lowest activity)

The definition of pixel activity and a proper threshold setting guarantee that only low activity pixels will be classified into low category. In Figure 3.5, the green rectangular is a partially-textured macroblock. Each black block represents a pixel. The blocks with blue, red or yellow contours are overlapped blocks, whose sizes are 4x4. Two pixels are marked with dots. The activity of the blue pixel is equivalent to the activity of the blue overlapped block, and the red pixel activity is equal to the activity of the red block.

The blue block has only smooth content and its activity should be low enough for the blue pixel to be selected into the low category. In contrast, all overlapped blocks containing the red pixel have some textured content. The red block has the lowest activity, but it is still textured, decreasing the possibility for the red pixel to end up in the low category.

Chapter 4 Distortion Compensation Algorithm

In chapter 3, target pixels were found. When coding an image, these pixels suffer from high MB QP. Therefore, a distortion compensation algorithm is proposed to improve these pixels. The algorithm is introduced in 4.2. Prior to it, a theoretical analysis together with a series of experiments are presented to demonstrate how the distortion function in the Lagrangian function can be changed to improve target pixels.

4.1 Lambda experiments

4.1.1 QP versus Lambda

The Lagrangian multiplier λ is a function of QP. Both λ and QP control the trade-off between bit cost and fidelity, but in different ways:

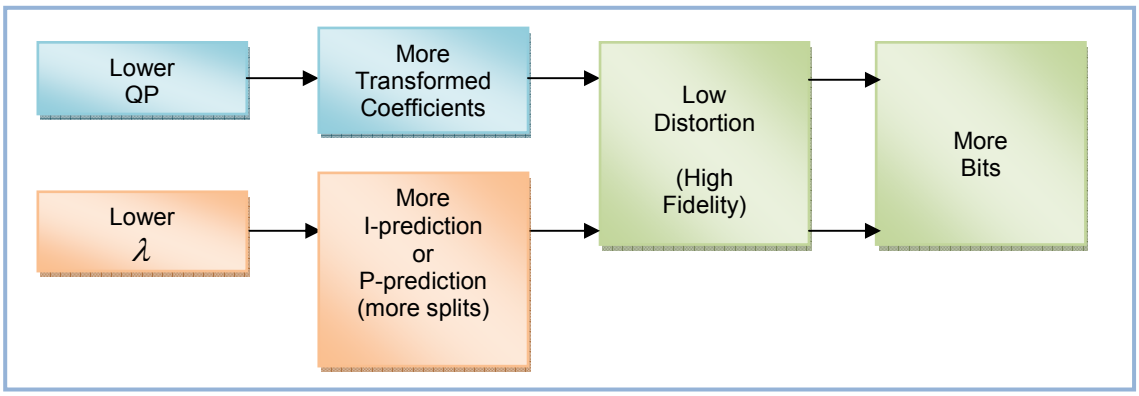
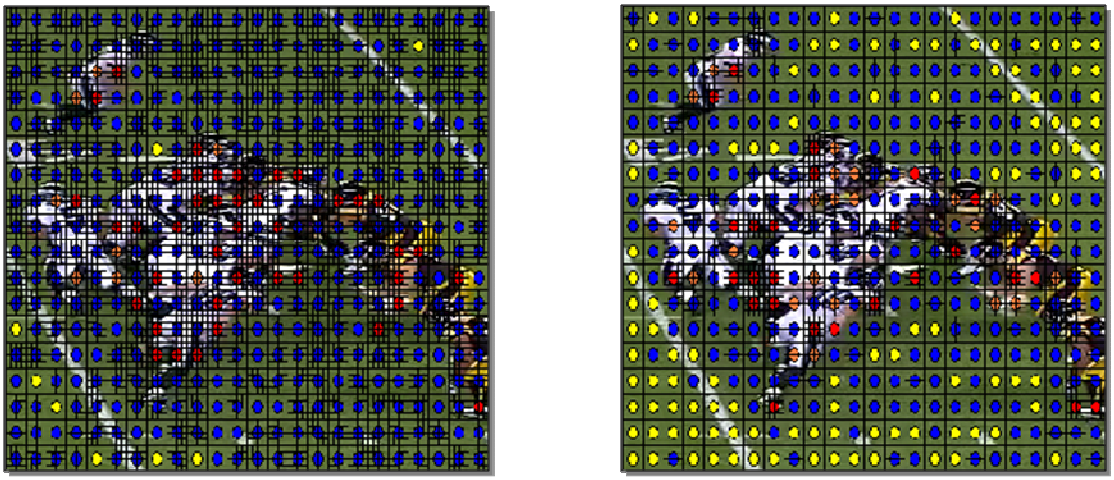


Figure 4.1 *Different methods to balance bit cost and fidelity*

Encoding a macroblock with lower QP generates less distortion (see section 1.4.8). Decreasing λ will make the RDO choose encoding modes (such as I-prediction or P-prediction with more and smaller splits) that provides high fidelity (see Figure 4.2 and 1.4.4). Therefore, changing λ has a similar effect on distortion improvements as changing QP.

4.1.1.1 Experiment 1: Objective comparison

To verify that changing λ has a similar effect to changing QP, an experiment was done, using NBC_Clip6.avi as the test sequence. To decrease λ, a lambda multiplier “c” is added in the Lagrangian function: $J = D + c \times \lambda R$. For each cλ, multiple fixed base QPs are tested and listed in table 4.1.



(a) c =0.06 *(b) Normal, c =1*

Figure 4.2 *More splits and intra-prediction modes (red) are chosen in (a), comparing to (b). Yellow dots represent skip mode. Blue dots represent inter-prediction modes. Red and orange dots represent intra-prediction modes.*
(Pictures are generated by Elecard Streameye V.3.0)

c	Base QPs			
1(anchor)	33	34	36	37
0.6	33	34	36	38
0.4	33	34	36	38
0.25	33	34	36	38
0.16	34	36	38	40
0.10	36	38	40	42
0.06	36	38	40	42

Table 4.1 Experiment parameters

Rate-distortion performance curves are plotted in Figure 4.3. The distortion is measured by PSNR. Taking QP=36 for example, Figure 4.4 shows that the lower λ used, the higher PSNR is obtained. It proves that decreasing λ can improve the compressed video quality (decrease distortion), similar to decreasing the QP.

Though changing λ brings good results, only small changes (representing small QP changes) are acceptable. As shown in Figure 4.3, when c equals to 0.10 or 0.06, the compression performance is much worse compared to the anchor. When λ gets smaller, the bit rate increases at an increasing rate, but the PSNR increases at a decreasing rate (Figure 4.4). Therefore large λ changes are not compression efficient.

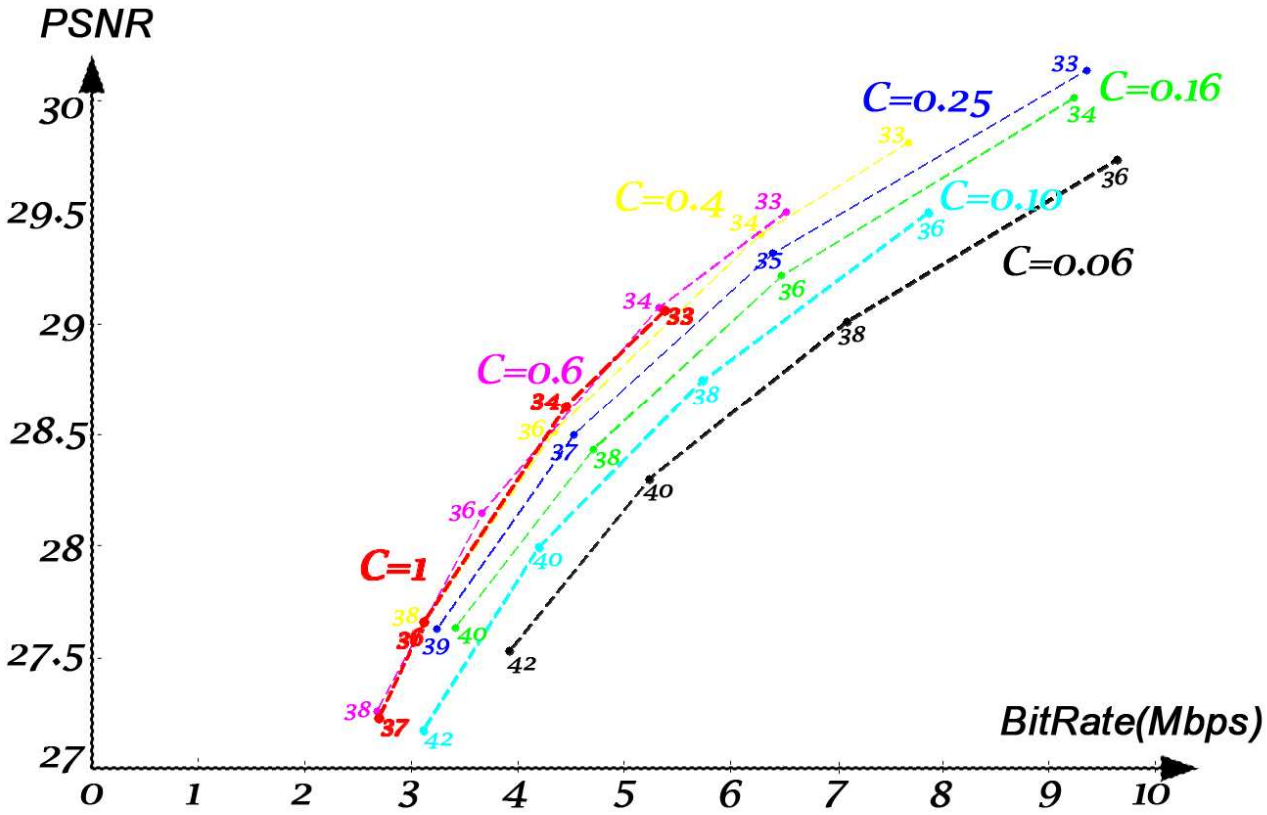


Figure 4.3 *Rate-distortion performance curves*

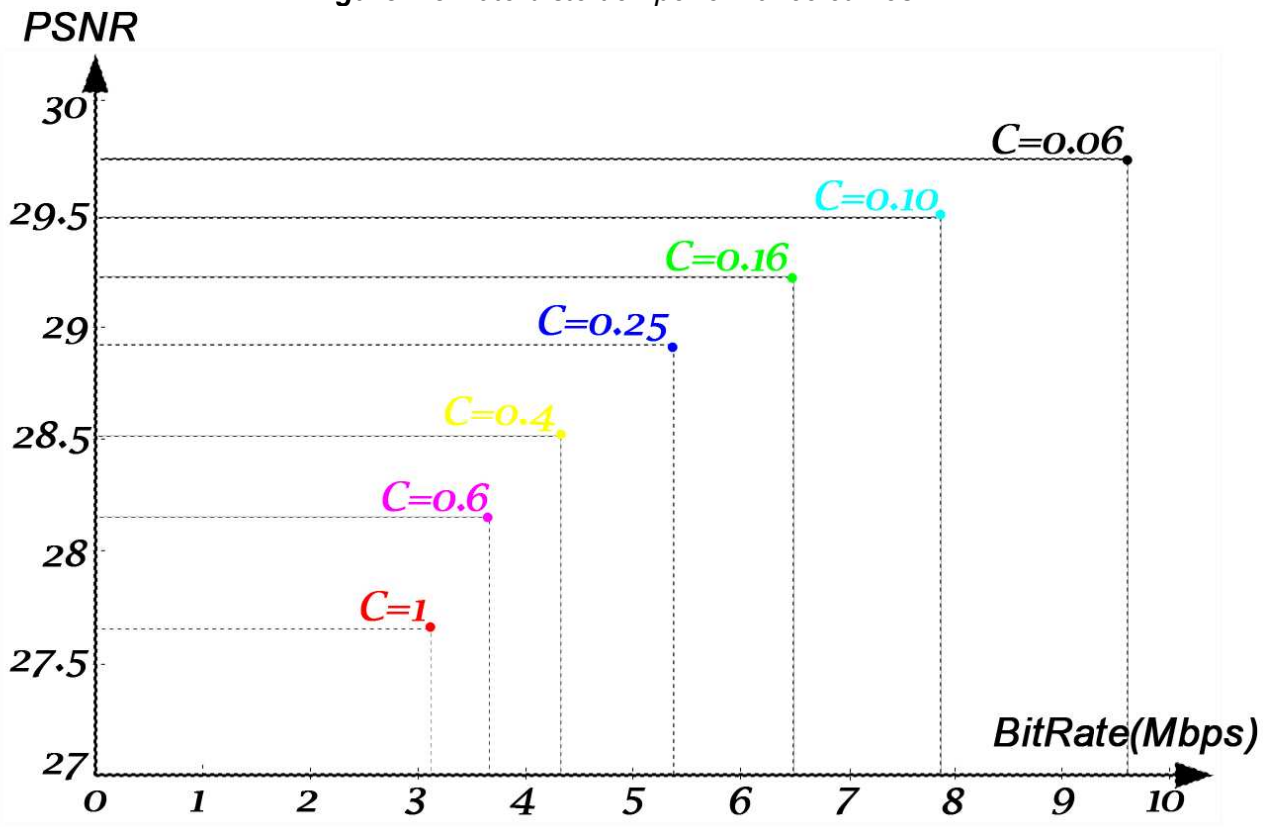


Figure 4.4 *Comparing performances when encoding with different lambda multipliers*

4.1.1.2 Experiment 2: Subjective comparison

In this experiment, subjective comparisons were made to observe different effects on distortion improvements caused by either lowering the λ or decreasing the QP. 72 macroblocks which suffer from serious ringing artifacts, are selected (see Figure 4.5). Two methods were tested to reduce the amount of ringing artifacts:

- Case 1: The 72 macroblocks are forced into low category, assigned with lower QP.
- Case 2: Use a lower λ when encoding these 72 macroblocks;
  Tested λ multipliers c = {1 (normal), 0.6, 0.4, 0.25, 0.16, 0.1, 0.06}

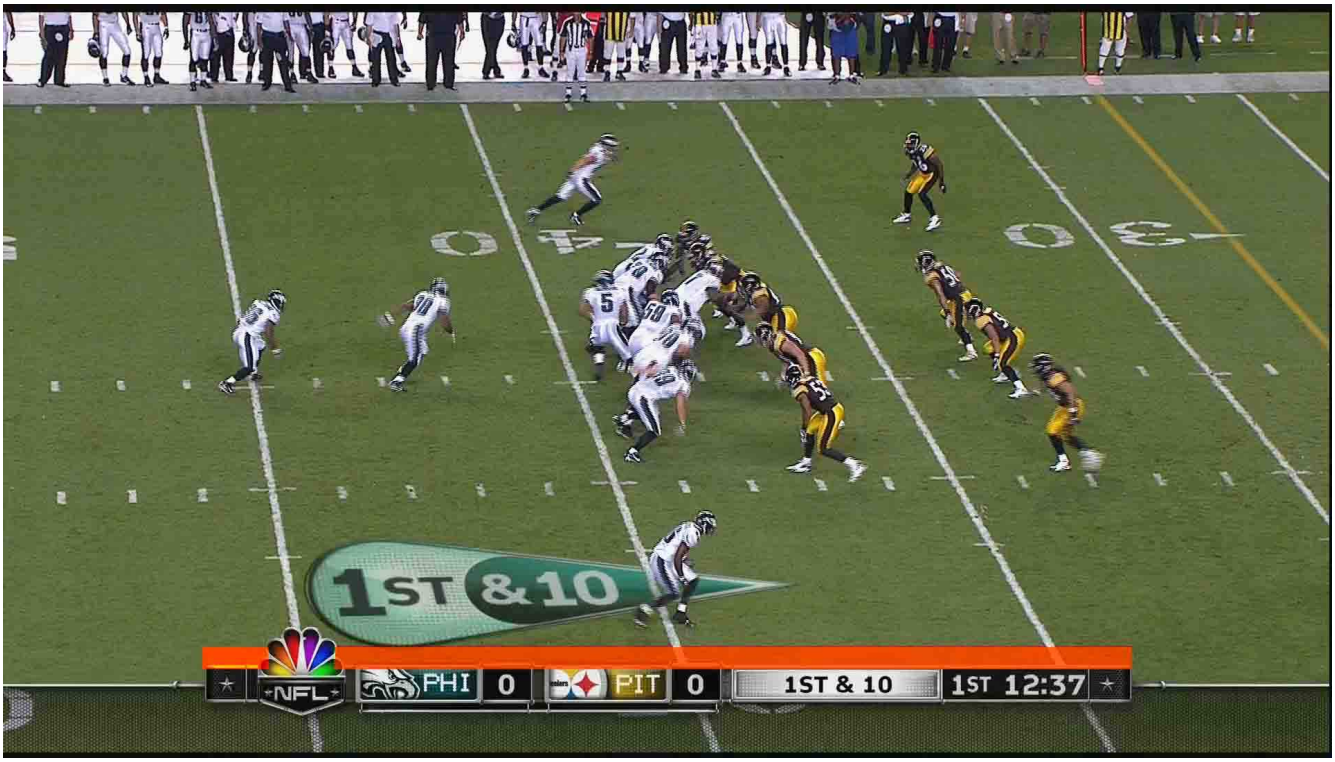


Figure 4.5 *72hard coded macroblocks (marked in orange)*

The result was subjectively evaluated in Ericsson research multimedia lab. Decoded video clips were displayed on a Sony 1920x1080 HDTV. The clips were carefully observed at a close distance (<2m).

Compared to normal encoding, quality improvements in the 72 macroblocks were observed for both methods. The effect of lowering QP is subjectively similar to lowing λ to 0.25λ. The 0.06λ gives the least distortion, but at the cost of the highest bit consumption.

To summarize, objective and subjective experiments both prove that changing λ has a similar effect as a QP change.

4.1.2 Lambda versus Distortion

It is not hard to see that changing λ is equivalent to change the distortion D. For instance, let $J_1 = D + \frac{1}{k} \times \lambda R$ and $J_2 = k \times D + \lambda R$, where k is denoted as “distortion multiplier”. Though J_1 =

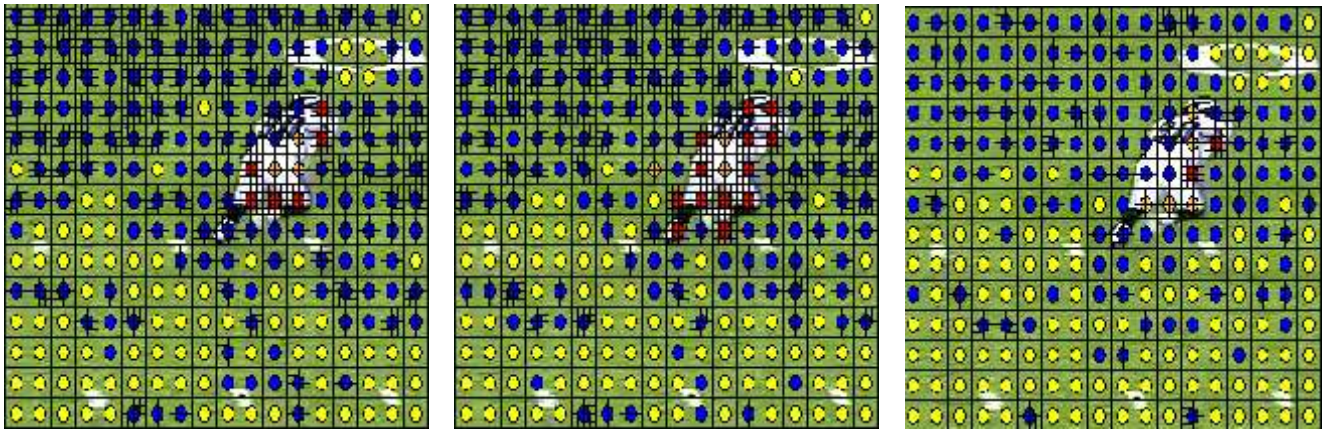
kJ_2, the finally chosen mode will be same, since the Lagrangian costs of all modes are multiplied with k in J2, giving no effects on mode decision. Therefore, if lowering λ can be a way to improve CODEC performance, so does increasing the D. Encoding with either method, more intra-prediction modes are chosen and more macroblock splits are obtained.

4.1.2.1 Experiment 3: Distortion

To verify that the increment of D gives a similar effect as decreasing the λ, Lagrangian functions were modified:

- Case 1: J = D + 0.25 x λR;
- Case 2: J = ($4xD_{Luma}$ + D_{Chroma}) + λR;

In case 2, only luma D has been multiplied with 4.



(a) λ Change *(b) Distortion Change* *(c) Normal Reconstructed*

Figure 4.6 *Comparing effects of changing distortion with changing λ.(a) gives similar result as (b). Comparing with (c), both (a) and (b) have chosen more modes which provide high compression quality. Yellow dots represent skip modes. Blue dots represent inter-prediction modes. Red and orange dots represent intra-prediction modes.*
(Pictures are generated by Elecard Streameye V.3.0)

Comparing (a) and (b) in Figure 4.6, it is shown that changing distortion has similar effect to λ change. In contrast with (c), both (a) and (b) have chosen more modes that provide comparatively high encoding quality. The difference between (a) and (b) is caused by the fact that the chroma distortion is accounted for in case 1 but not in case 2.

4.2 Distortion Multiplier and Distortion Compensation Algorithm

In the previous section, it was shown that changing the Lagrangian multiplier λ has a similar effect as changing QP. It was also shown that changing distortion D is equivalent to changing λ. Adjusting either one of these three factors (QP, λ and D) can improve the distortion on macroblock level at the expense of higher bit cost.

In this thesis, only target pixels should be improved. Therefore, partitioning QP, λ or D into pixel level is required. In chapter 2, the conclusion was drawn that changing QP within a macroblock is out of the question. Though partition of λ on a sub-macroblock level may work,

splitting the bit cost R is a tough task, so λR is hard to be calculated for individual pixels. To get rid of these problems, partition of the distortion D is proposed.

First of all, a distortion multiplier k is introduced as below:

SSD is the most used measure of distortion in RDO. A modified distortion function is:

$$J = D + \lambda R = (D_{luma} + D_{chroma}) + \lambda R$$

$$D_{luma}^{A} = \sum_{(x,y)\in A} \left| r_{x,y} - s_{x,y} \right|^2 \times k \qquad \begin{cases} k \geq 1, & if\ r_{x,y} is\ a\ target\ pixel \\ k = 1, & otherwise \end{cases} \qquad (4.1)$$

where k is the distortion multiplier. A is a set of low activity pixels of the current macroblock (or a split of the current macroblock). r and s denote a reconstructed pixel and a source pixel respectively. Since chroma distortion is comparatively small as mentioned in section 1.3.3, only luma distortion is changed in this thesis.

Lowering the λ is equivalent to increasing the k. For a target pixel, k should be bigger than 1, so that the distortion term of this pixel is increased.

4.3 Thesis Algorithm

A summarization is presented here to give a general and clear view of the whole algorithm --- the combination of the proposed pixel classification and distortion compensation algorithms.

The thesis algorithm takes place in between the adaptive QP section and the mode selection section as seen in Figure 4.7.

For macroblocks of the current frame:
- The adaptive QP algorithm will categorize macroblocks in this frame.
- Low activity macroblocks are assigned a lower QP, while high activity macroblocks are assigned a higher QP.

For each pixel, pixel classification is done:
- Calculate the pixel activity.
- Sort pixel based on pixel activities.
- Pick out target pixels.

For each target pixel, the distortion compensation factor is computed:
- Calculate $k_{x,y}$ due to formula (4.2)

For the current macroblock, select a mode for prediction
- Calculate the Lagrangian cost function with the new luma distortion formula (4.1).

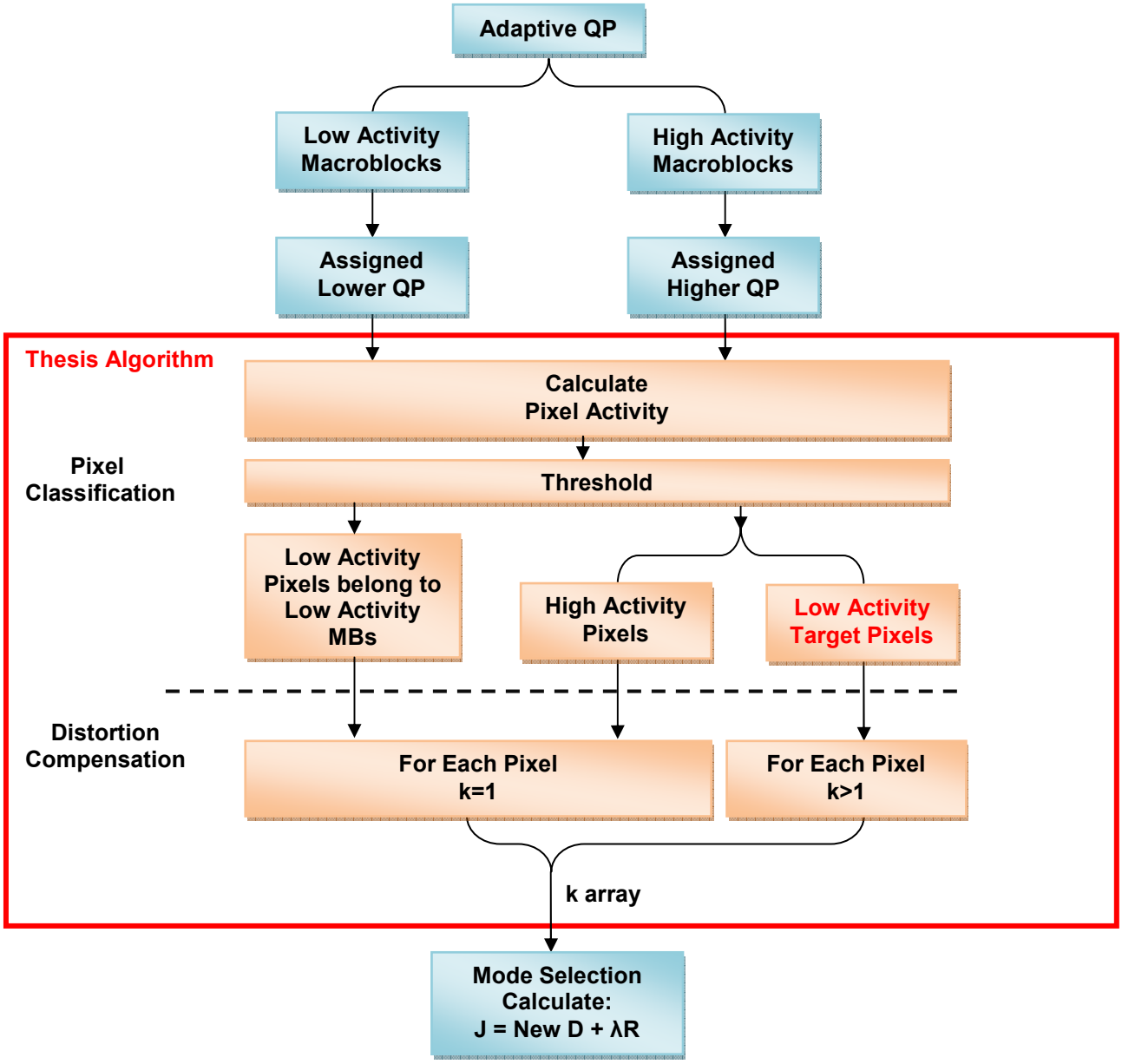


Figure 4.7 *Flow Chart of the whole algorithm*

Chapter 5 Simulations

In previous chapters it is mentioned that the size of overlapped blocks, the threshold controlling the number of processed pixels and the distortion multiplier k are manually set before encoding. In this chapter, simulation results are presented. First good combinations of blocks sizes and threshold settings are found. Secondly, different values of distortion multiplier k are tested at various bit rates.

5.1 Optimizing the Classification Algorithm

To optimize the classification algorithm, the size of overlapped blocks, low activity categories and the pixel activity threshold need to be carefully specified. They may greatly influence the subjective quality:

- **Size of overlapped blocks**

The size of overlapped blocks is related to pixel activity and the classification result. Large overlapped blocks result in higher pixel activity and vice versa. Varied overlapped block sizes lead to different classification results. For example, with the overlapped block size of 4x4, a pixel may be specified as low activity pixel. However, if use 16x16 instead, the pixel activity may increase over the threshold pixel activity, so that the pixel is classified as high activity pixel. Only low activity target pixels will have their RDO distortion scaled with a factor larger than 1.

- **Low activity categories**

In practice, more than two activity categories are used during the macroblock adaptive QP procedure. The low activity category should be specified, since pixels belonging to low activity macroblocks will be assigned a distortion multiplier k equal to 1.

- **The pixel activity threshold**

The threshold decides the number of low activity pixels. With a proper threshold, the classification algorithm is being prevented from choosing pixels of high textured content.

The following simulations in this section are presented to demonstrate how these three parameters are set.

5.1.1 Optimizing the Size of Overlapped Blocks

The optimal block size is dependent on specific scene content and particular encoding requirements. E.g. the primary requirement of this thesis is to suppress the ringing artifacts around textured figures against a smooth background. Therefore, a block size that results in lower activities for pixels which are close to figures, but not a part of them is needed. In this thesis, the tested block sizes are: 16x16, 8x8 and 4x4. For each test, the first 65% lowest activity pixels are plotted in blue in Figure 5.1 to 5.4. Comparisons and conclusions are made in the end.

- **Test 1-1: Evaluating overlapped blocks of size 16x16**
- Test sequence: NBC_Clip6.avi Threshold: 65 **Block size: 16x16**

Figure 5.1 *The first 65% lowest activity pixels are dyed in blue (overlapped block size: **16x16**)*

- **Test 1-2: Evaluating overlapped blocks of size 8x8**
- Test sequence: NBC_Clip6.avi Threshold: 65 **Block size: 8x8**

Figure 5.2 *The first 65% lowest activity pixels are dyed in blue (overlapped block size: **8x8**)*

- **Test 1-3: Evaluating overlapped blocks of size 4x4**
- Test sequence: NBC_Clip6.avi Threshold: 65 **Block size: 4x4**

Figure 5.3 *The first 65% lowest activity pixels are dyed in blue (overlapped block size: **4x4**)*

- **Conclusion:**

For the NBC_Clip6 sequence, the optimal size of overlapped blocks is 8x8. This decision is based on careful comparisons between Figures 5.1, 5.2 and 5.3. A zoomed-in part each figure are presented here (Figure 5.4) to demonstrate why 8x8 is the optimal one.

(a)16x16 *(b)8x8* *(c)4x4*

(d)16x16 *(e)8x8* *(f)4x4*

Figure 5.4 *Details of Figures 5.1, 5.2 and 5.3.*
Particular attention is paid on picture content included in red circles.

Observing (a) (b) and (c), a 16x16 block size is too big to make pixels above the scoreboard chosen, while a block size of 4x4 selects many pixels of the textured part, which will cost unacceptable large number of bits. A block size of 8x8 is a compromise between (a) and (c) in that pixels above the scoreboard are selected and there aren't many pixels chosen in the margin or inside figures. In (e), unselected pixels exist in interspaces between players, while in (f), a few blue pixels in players (or lines) are undesired. Though (e) may not be the perfect one, it is the preferred choice.

5.1.2 Specifying Low Activity Categories

In the existing adaptive QP algorithm, macroblocks are classified into four categories according to macroblock activity. Each category is assigned a different DQ:

$$\begin{cases} \text{Category 0} = -2DQ; \\ \text{Category 1} = -DQ; \\ \text{Category 2} = 0; \\ \text{Category 3} = +DQ; \end{cases}$$

Macroblocks in category 0 have the lowest activities, whereas those in category 3 have the highest activities. In Figure 5.5-5.9, target pixels are plotted in blue. Except macroblocks in category 0, they are coloured according to the categorization.

Category 0: Original colour;
Category 1: Green;
Category 2: Pink;
Category 3: Red.

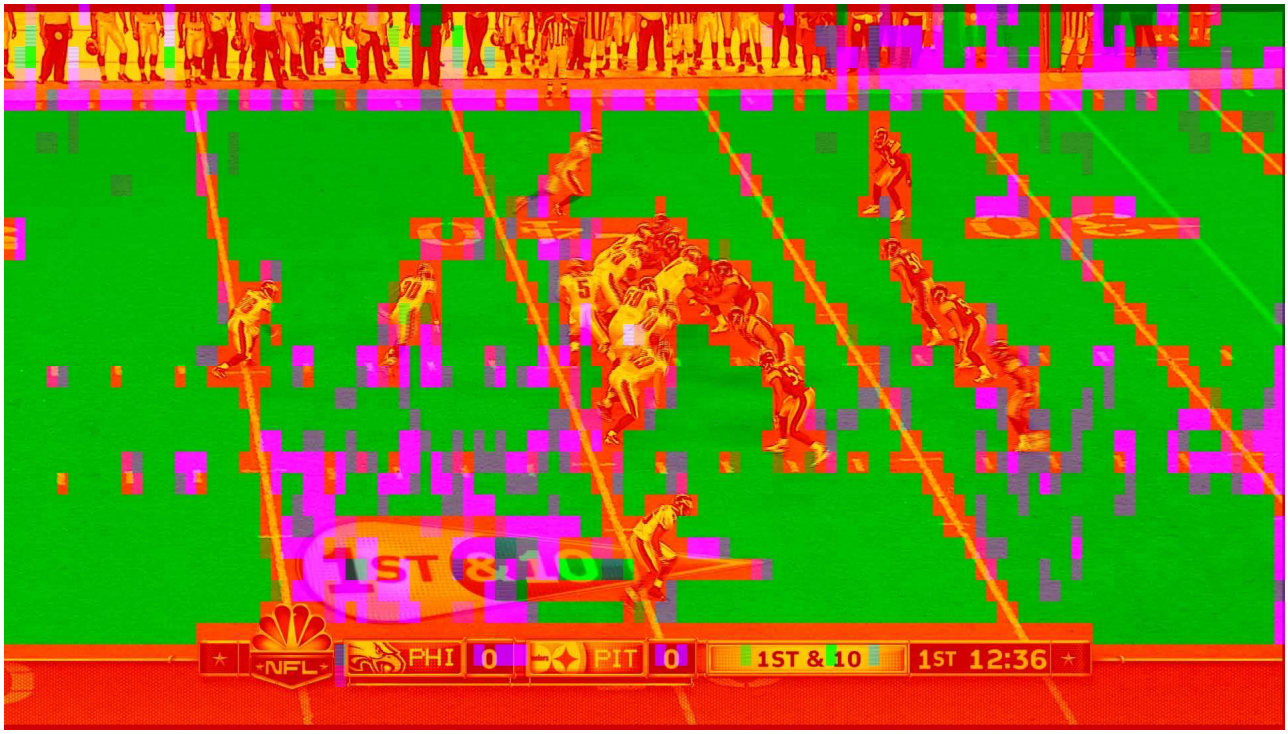


Figure 5.5 *Category map.*

Since there are four categories, the low activity categories should be specified, in which pixels are assigned a distortion scaling factor K of 1. Category 0 and 1, where the distortions are improved by lower QPs (negative DQs), are specified as low activity categories. There are both red and pink macroblocks located at where ringing artifacts may appear, despite that some pink ones have smooth content. Subjective tests are done to decide whether category 2 (pink) should be specified as one of the low activity categories as shown in Figure 5.6-5.8.

The threshold set in the following tests has the same meaning as in previous tests. For instance, a threshold of 65, means that the first 65% lowest activity pixels will be chosen. In this case, these 65% pixels are made up by both pixels in low activity categories and target pixels.

- **Test 2-1: Specifying Low Activity Categories**
- Test sequence: NBC_Clip6.avi Threshold: 65 Overlapped block size: 8x8
  Category 0 and 1 are considered as the low activity categories;
  Category 2 and 3 are considered as high activity categories;

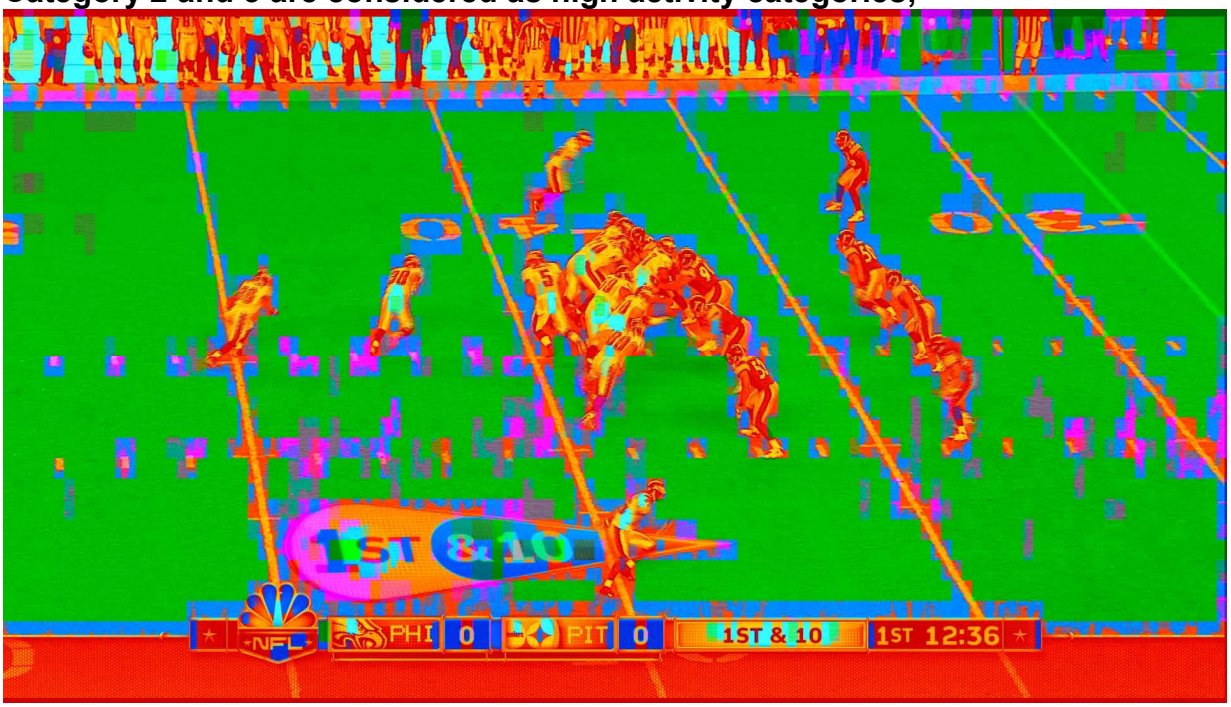


Figure 5.6 *Choose target pixel (blue) from macroblocks in category 2 and 3;*

- **Test 2-2: Specifying Low Activity Categories**
- Test sequence: NBC_Clip6.avi Threshold: 65 Overlapped block size: 8x8
  Category 0, 1, 2 are considered as the low activity categories;
  Category 3 is considered as the high activity category;

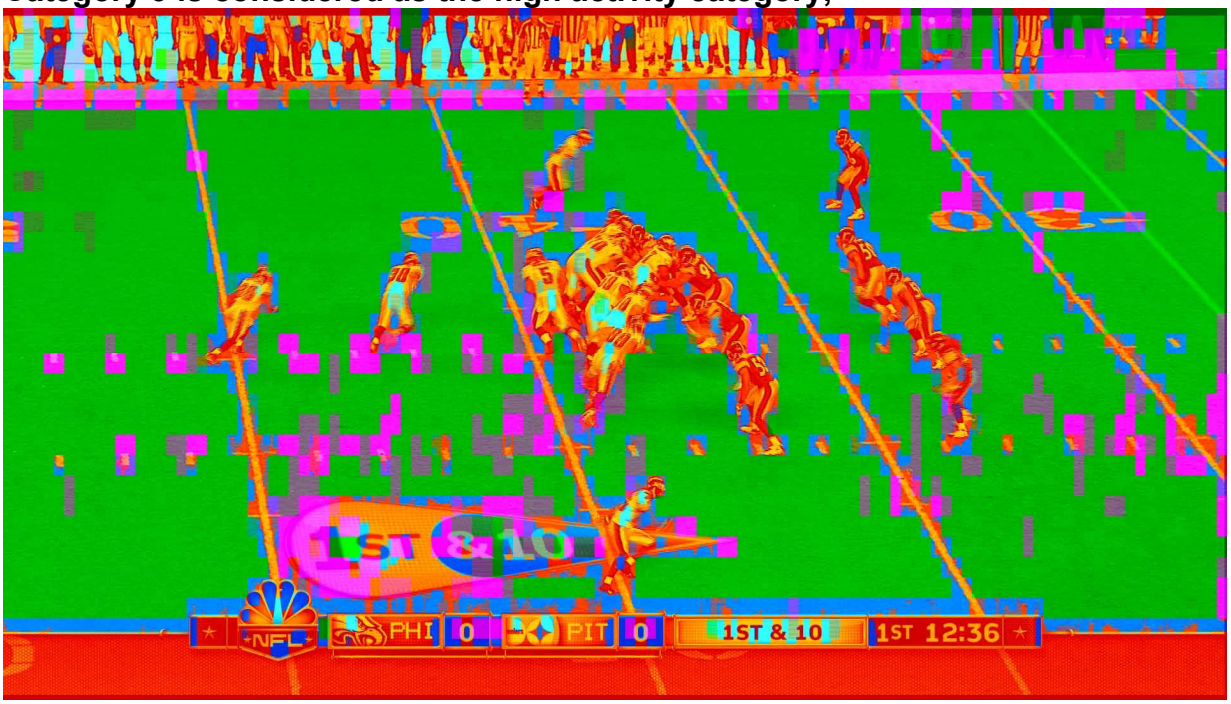
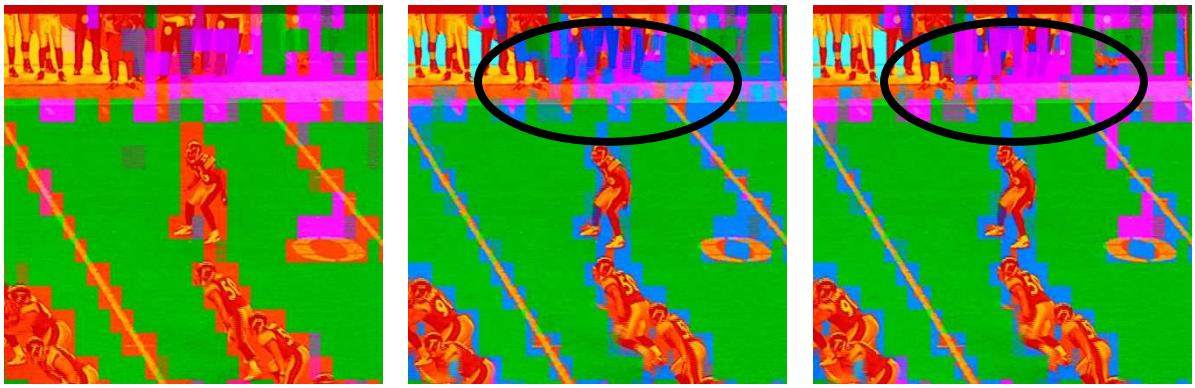


Figure 5.7 *Choose target pixels (blue) from macroblocks in category 3;*

- **Conclusion:**

For the NBC_Clip6 sequence, it is better to include category 2 into the group of low activity categories. That is, pick out target pixels only from the macroblocks in category 3. Zoomed-in parts are shown in Figure 5.8.



(a) Category Map *(b) Category 2 and 3* *(c) Category 3*

Figure 5.8 *Contrastive slices cut from Figure 5.5, 5.6 and 5.7. Particular attention is paid on picture content included in black circles.*

In contrast with (b), (c) shows that selecting no pixels from category 2 can effectively avoid choosing pixels of smooth content, (e.g. the audience) and decrease the total number of target pixels to prevent unnecessary bit costs.

5.1.3 Optimizing the Threshold

As presented in the category map (Figure 5.5), category 3 contains macroblocks that have high textured content, (e.g. players, audience, grids below the scoreboard, and so on.). Therefore the threshold should be set very carefully so that target pixels can be selected properly, also making sure that no pixel in high textured macroblocks is chosen. Several values are tested (Threshold = 55, 60, 65, 70, or 75) and pictures similar to Figure 5.7 are plotted in appendix B.

- **Conclusion:**

Based on the full test presented in appendix B, the optimal threshold for the NBC_Clip6 sequence is 70. The target pixels then almost cover the whole row above the scoreboard, the areas along all lines on the ground and fully around players, as shown in Figure 5.9.

- **Test 3: Find the Optimal Threshold**
- Test sequence: NBC_Clip6.avi **Threshold: 70** Overlapped block size: 8x8
  Category 2 is considered as low activity category;

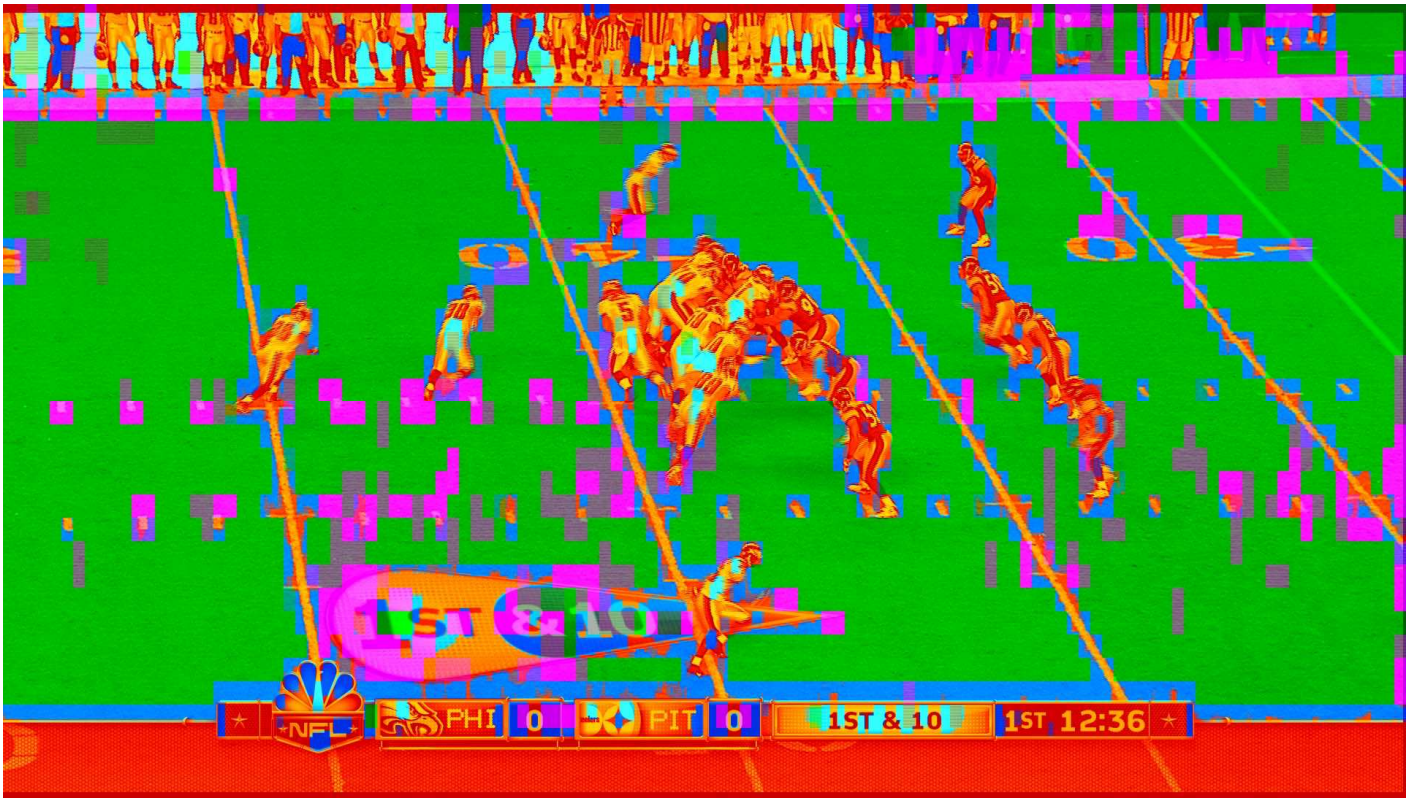


Figure 5.9 *Choose target pixels (blue) when the threshold is set to 70.*

5.2 The Distortion Multiplier k

The preferred block size (8x8), and the preferred threshold (70%) have now been chosen. With these parameters, different values of the distortion multiplier k (k =1, 4, 8, 16 or 32) have been tested under various bit rates (5.0 Mbps and 8.0 Mbps). k=1 is the anchor, which is used for comparing rests of this test. Objective and subjective assessments are presented here.

5.2.1 Objective Assessment

The objective performance is getting worse as k increases. In the anchor, the Lagrangian cost function chooses optimal modes for the given QP to keep the distortion-rate balance. When k>1, the modified Lagrangian cost function weights more on distortion, resulting in choosing more expensive modes at an increasing bit-cost. To keep the balance, the rate-control needs to increase the base QP to keep the bit rate, so that PSNR decreases as presented in Table 5.1 and 5.2.

	PSNR(dB luma)	Bit Rate (Mbps)
k=1 (the anchor)	29.053	5.330
k=4	28.934	5.566
k=8	28.794	5.566
k=16	28.702	5.361
k=32	28.159	5.368

Table 5.1 *Compare objective performance when bit rate is 5.0 Mbps.*

	PSNR(dB luma)	Bit Rate (Mbps)
k=1 (the anchor)	30.454	8.641
k=4	30.434	8.612
k=8	30.320	8.643
k=16	30.083	8.622
k=32	29.759	8.611

Table 5.2 *Compare objective performance when bit rate is 8.0.*

This objective result was as expected. Since PSNR cannot perfectly represent human perception of distortion, subjective assessment is the main concern in this thesis.

5.2.2 Subjective Assessment

Subjective assessment is carried out in the Ericsson research multimedia lab. NBC_Clip6 sequences coded with different k are displayed on a Sony 1920x1080 HDTV. Comparisons are made with the anchor, by careful observations at a close distance in front of the TV (<2m).

For the NBC_Clip6 sequence, the effect of the distortion compensation algorithm is shown in Figure 5.10. Quality improvement can be observed mainly in red coloured areas, while distortion increases in blue marked regions. In the anchor, the ringing artifacts are most serious above the scoreboard and around players. Those areas are the main focus when comparing the sequences.

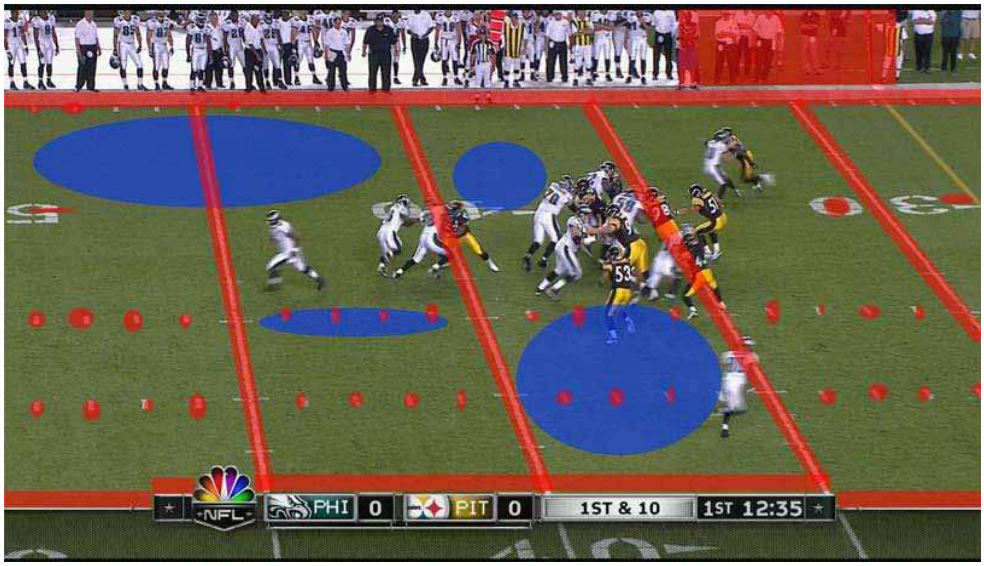


Figure 5.10 *Subjective quality is improved in red areas, while it is worse in blue areas. (Frame Number = 40)*

- **Simulations at bit rate 5.0 Mbps**
- k=4
  In contrast with the anchor, the distortion has been suppressed sightly. Ringing artifacts around the scoreboard are still present. No noticeable improvements are seen around players.

- k=8
  Obvious quality improvements are found on the row of macroblocks above the scoreboard. However, the effect of distortion compensation gets weak since the base QP gets higher to compensate for an increased bit cost. Few macroblocks around players has been improved in quality. Distortion is observed on the grass.

- k=16
  The ringing artifacts around scoreboard have been reduced even more. For the distortion around players, in some areas it has been compensated, but in other areas it gets worse. The total amount of ringing artifacts around players hasn't changed much. Due to the increment of the base QP, serious distortion appears on the grass (in the blue areas of Figure 5.10).

- k=32
  k=32 has a similar effect as k16. However, as the base QP increases, serious blurring and shake artifacts appear on the background.

5.2.3 Simulation Conclusions

- **Goal Achievement**

For the NBC_Clip6 sequence, ringing artifacts above the scoreboard are improved effectively. As shown in Figure 5.11, the same region (a part of the scoreboard) is shown from each decoded sequence and the original uncompressed sequence. Residuals are obtained by subtracting the original pixel values from each reconstruction.

The residuals show the effect of distortion compensation. As seen, the residual of the anchor, obviously contain more artifacts than the other residuals.

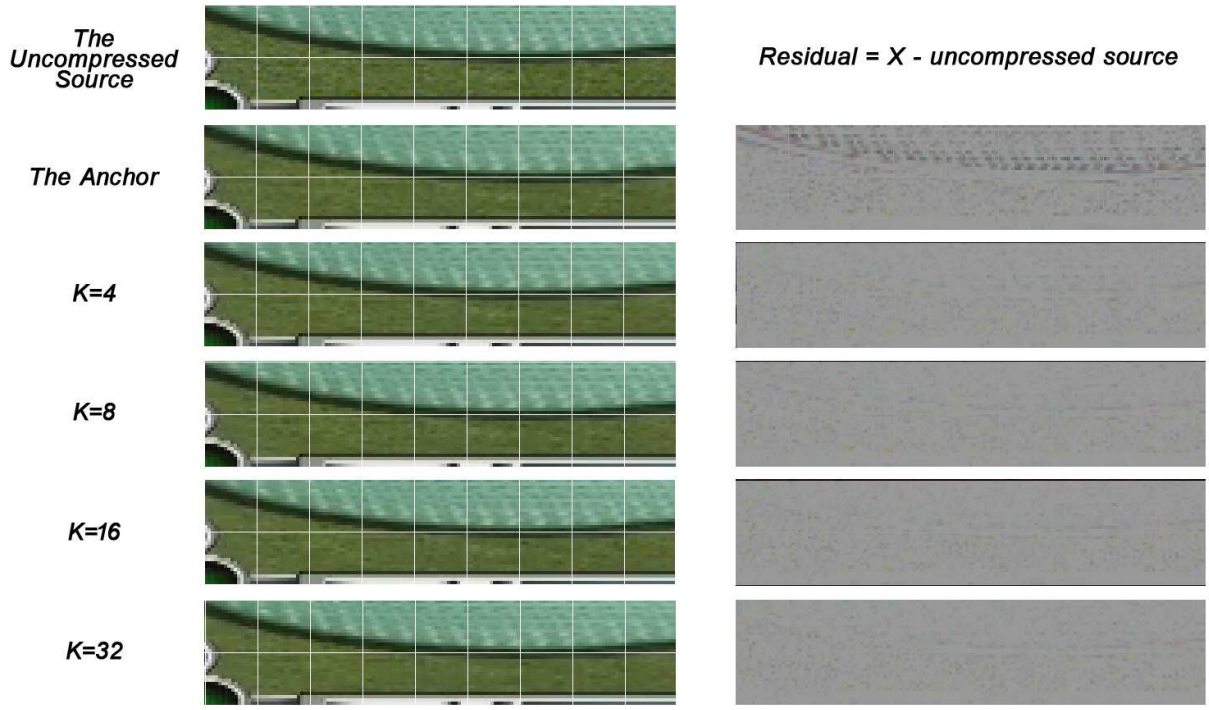


Figure 5.11 *Subjective evaluation of the effect of the distortion compensation algorithm. (Each slice shows 18 macroblocks above the scoreboard.)*

- **Algorithm Efficiency**

It is hard to discern the difference between residuals corresponding to k=4, 8, 16, and 32 in Figure 5.11. But residual of k=4 is indeed larger than the residual of k=8, 16 or 32. However, the residuals corresponding to k=8, 16 and 32 are almost the same. This is explained by Figure 5.12.

The smallest split of a macroblock is a 4x4 block, which gives the most accurate prediction (Refer to 1.4.6). In contrast to the anchor, k=4, 8, 16, or 32 have more splits for macroblocks which contain target pixels, resulting in a better prediction and less residual shown in Figure 5.11. From k=4 to k=8, it is seen that the number of splits increases, while from k=8 to k=16 and k=32, there are no evident increment of splits. A distortion compensation with a k higher than 16 is not very effective.

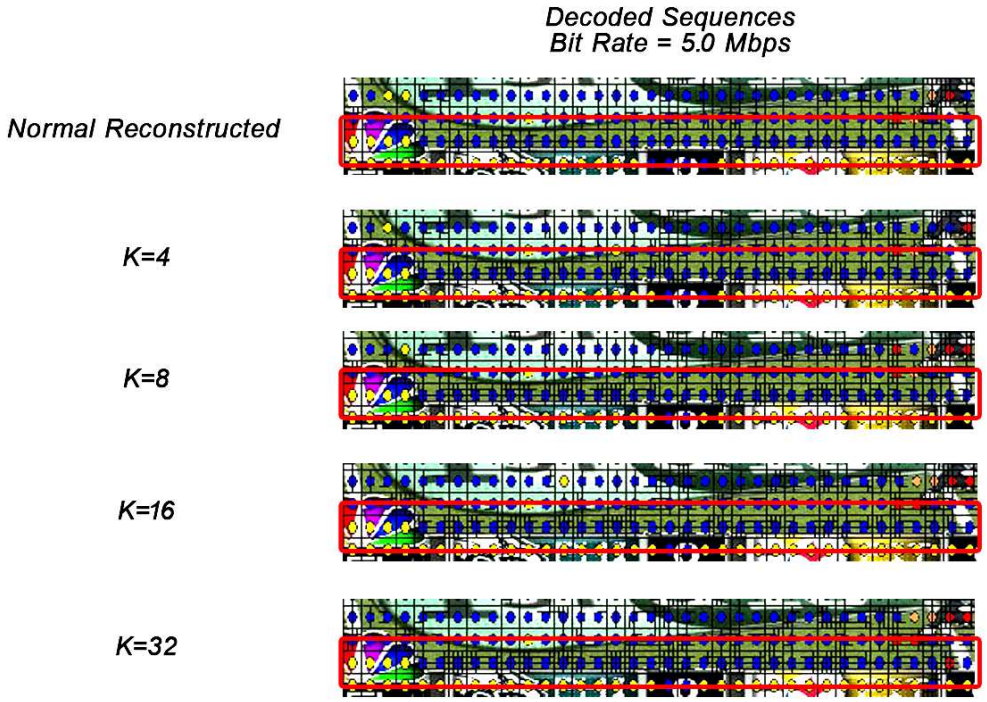


Figure 5.12 *Analyse what modes are used to predict macroblocks.*
The picture is generated by Elecard StreamEye v3.0.
(Each slice shows macroblocks above the scoreboard.)

- **Simulations at Higher Bit Rate**

When the bit rate is increased to 8.0 Mbps, lower QPs are used during encoding. The distortion scaling factors k=1, 4, 8, 16 and 32 are used. In contrast to 5.0 Mbps encoding, the distortion compensation algorithm is now less effective.

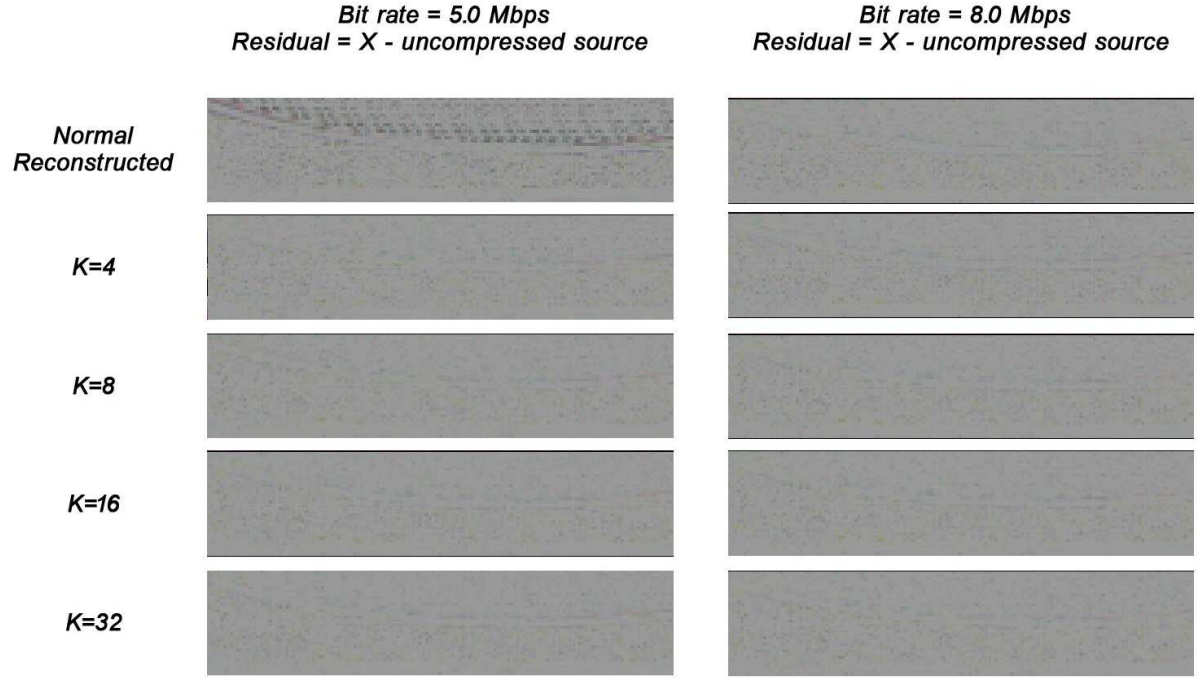


Figure 5.13 *Comparison of the residuals between sequences coded at bit rate 5.0 and 8.0 Mbps*

As shown in Figure 5.13, the residual, which corresponds to the anchor sequence coded at bit rate 8.0 Mbps, contains less information than the one coded at bit rate 5.0. Since the average quality is improved by using lower QP, the distortion compensation algorithm appears to be less effective.

- **Influence of Increasing Base QP**

With the distortion compensation algorithm, the Lagrangian cost function chooses more expensive modes to suppress ringing artifacts. The rate control needs keep the bit cost, resulting in an increasing base QP and a decreasing average coded quality.

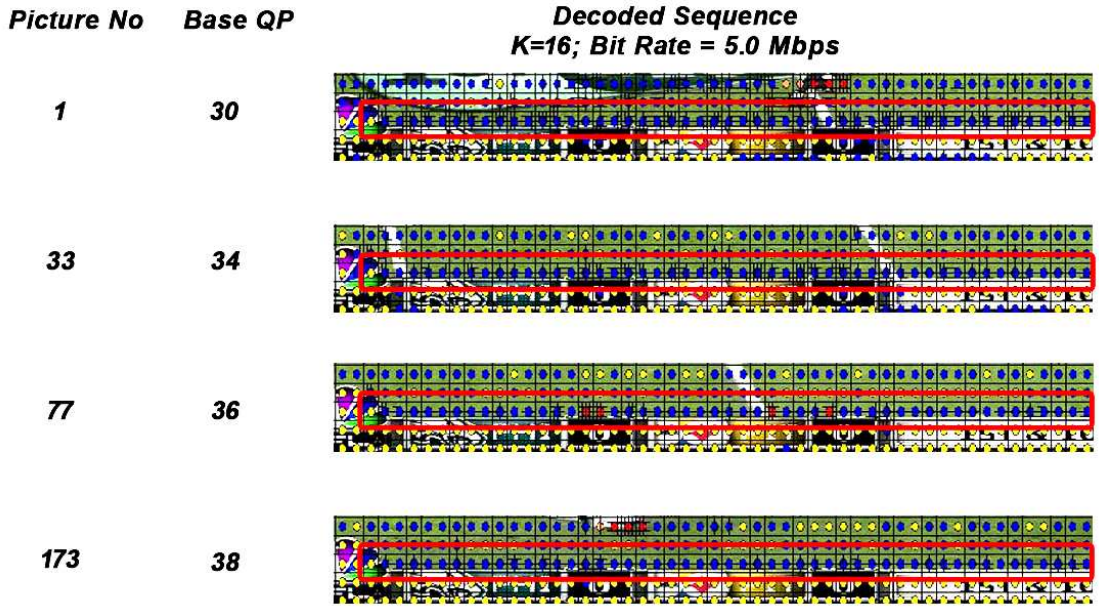


Figure 5.14 *The result of increasing base QP.*
The effect of the distortion compensation algorithm decreases.
Less number of splits is found in macroblocks in the target area shown in red rectangles.
The picture is generated by Elecard StreamEye v3.0

The increasing Base QP also increases the Lagrangian multiplier λ, which counteracts the influence of the distortion compensation algorithm. For example, in Figure 5.14, when the base QP increases overtime, less and less splits are made, resulting in more and more serious unsuppressed ringing artifacts.

- **Problem of remaining ringing artifacts**

Quality improvements are also observed on some places around players but not all artifacts are removed. The distortion algorithm is not effective enough to compensate macroblocks around highly moving objects.

This problem is caused by two reasons:

1. Pixels around moving objects, such as the players have high pixel activities. But that doesn't mean macroblocks made up by these pixels always get proper distortion compensations. For instance, macroblocks marked with orange circles in Figure 5.15 haven't got any splits.

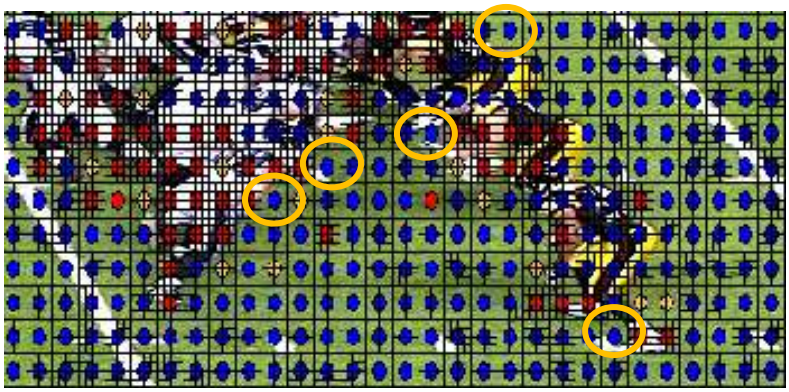


Figure 5.15 *Some macroblocks around players haven't got any splits. (K=8, Bit rate = 5.0 Mbps)*

A half textured macroblock may contain only a small amount of target pixels. The distortions are scaled for target pixels, but the SSD of the half textured macroblock may still not enough for choosing a better mode for prediction.

2. Macroblocks around highly moving objects are in the high activity category. Their MB QPs increases as the base QP increases. As shown in Figure 5.16, the average base QP of k=32 or k=16 sequences are higher than the anchor one. For macroblocks around players, though most of them get more splits from the distortion compensation algorithm (shown in Figure 5.15), they are still coded in low fidelity due to the high MB QP.

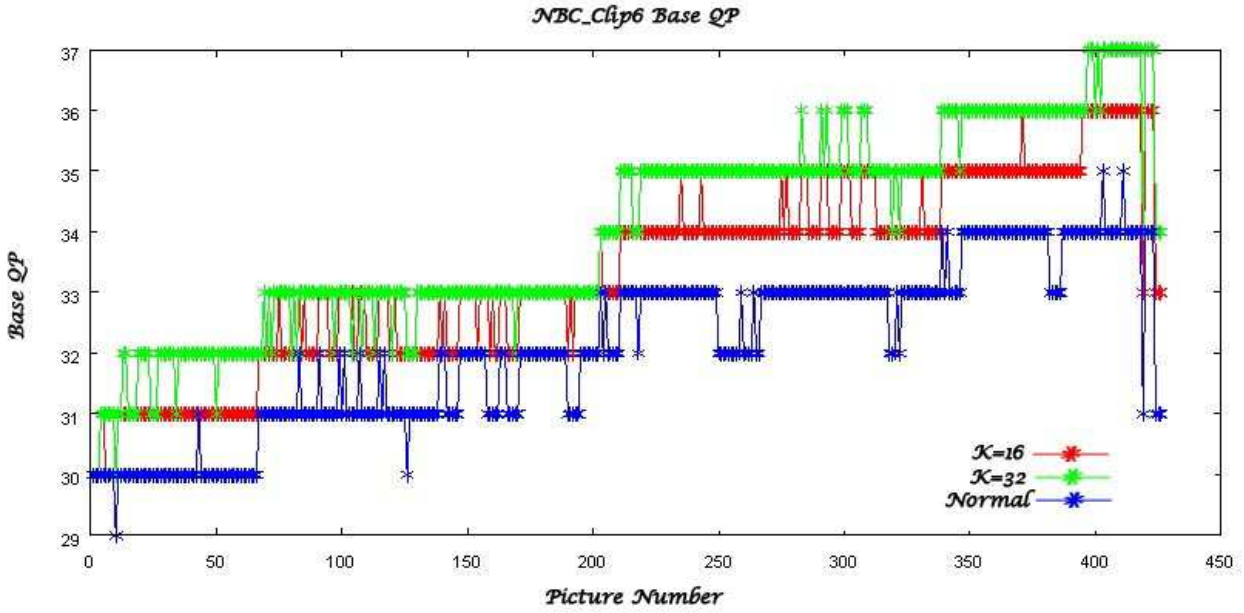


Figure 5.16 *Base QP increases as the picture number increasing.*

- **Optimize the Distortion Multiplier k**

As discussed previously, the distortion compensation algorithm is preferred to be applied when the bit rate is low. Considering the trade-off between algorithm efficiency, average coded quality, and distortion improvement, the distortion multiplier k of the NBC_Clip6 sequence should be set between 8 and 16. A k smaller than 8 is not effective enough and a k larger than 16 provide slightly improvements when comparing to the k of 16.

Chapter 6 Conclusion and Future Work

In chapter 6, the whole thesis work is summarized, further developments are discussed, and possible improvements of the algorithms are suggested.

6.1 Conclusion

In this thesis project, the video CODEC system and H.264 standard has been studied, as well as the rate-distortion theory. To achieve the primary goal of reducing ringing artifacts in coded video sequences, methods were designed to pick out target pixels and compensate them for distortion. The pixel classification algorithm was designed to find target pixels, and the distortion compensation algorithm was developed to suppress the ringing artifacts.

A feasibility analysis was done before the development of the algorithm to ensure that scaling of the distortion in RDO gives similar effects as lowering the QP. To optimize the classification algorithm, several important factors are proposed. The optimal overlapped block size, proper low categories specification and the optimal threshold are tested prior to real encoding. The NBC_Clip6 sequence is coded with different k settings. In consideration of compromising average quality and ringing artefact reduction, the distortion multiplier k between 8 and 16 is preferred. Performance improvement could not been seen when high Ks were applied.

Subjective assessment has been made. For the NBC_Clip6 sequence, ringing artifacts above the scoreboard is reduced. However the effect gradually decreases along with the increment of base QP. Higher k values increases the base QP even more. Distortion around players has not been suppressed effectively.

6.2 Future Work

Both the classification algorithm and distortion compensation algorithm can be improved in the future. For instance, now the distortion is compensated on target pixels in partially textured macroblocks. The distortions of target pixels are amplified and the total distortion of a frame is increased. A proposed idea is to compensate the target pixels, but meanwhile decrease the distortion of textured macroblocks covering no target pixels. The formula 4.1 can be changed to:

$$J = D + \lambda R$$

$$D_{x,y} = \sum_{(x,y)\in A} \left| r_{x,y} - s_{x,y} \right|^2 \square k \qquad (6.1)$$

$k > 1$, if $r_{x,y}$ is a target pixel;
$k = 1$, if $r_{x,y}$ is a pixel belonging to a low activity macroblock;
Use $\frac{1}{k}$, if $r_{x,y}$ is a high activity pixel belonging to a high activity macroblock which contains no target pixels.

By formula 6.1, the distortion of pixels in textured macroblocks decreases, so that less number of bits is needed for these macroblocks. It may be helpful to keep the bit rate and avoid incrementing of the base QP.

Furthermore, instead of distortion, partitioning lambda and/or bits can be an alternative. Then the bit cost can be a factor influencing the lambda multiplier and this might brings different effects and results.

In H.264, RDO is not only applied for mode decision alone, but also for selecting reference frames and motion vectors. Another idea is to implement the distortion compensation algorithm in the selection of motion vectors to provide more accurate motion estimations for macroblocks that includes target pixels.

What is proposed here is just a tip of an iceberg. There are still much more aspects that can be explored in the future.

References

[1] Iain E Richardson. H.264 and MPEG-4 Video Compression. September 2003, Wiley & Sons.

[2] Yu Zhaoming, Zha Riyong, Huang Lei, Zhou Haijiao. Video Coding Standard --- H.264. March 2006, Post and Telecom Press.

[3] Liu Feng. Video Coding Techniques and Standards. August, 2006, Beijing University of Post and Telecommunications.

[4] http://en.wikipedia.org/wiki/H.264

[5] Zhou JiongPan, Pang QinHua, Xu Dawo, Wu Weiling, Yang Hongwen. Communication Theory (3rd Edition). August 2008, Beijing University of Post and Telecommunications. p282-287.

[6] Ge XianXian, Wang Yu, Hao ChongYang, Yang LiNa. Rate-Distortion Optimized Strategy of H.264/AVC. October 2005, Wireless Communication Techniques Magazine.p14-18.

[7] Antonio Ortega, Kannan Ramchandran. Rate-Distortion Methods for Image and Video Compression. November 1998, IEEE Signal Processing Magazine. p23-49.

[8] Gary J. Sullivan, Thomas Wiegand. Rate-Distortion Optimization for Video Compression. November 1998, IEEE Signal Processing Magazine. p74-90.

[9] JVT of ITU-T AND ISO/IEC JTC 1 , Draft ITU-T Recommendation and Final Draft International Standard of JVT [S].

Appendix A: Lambda and QP

In this thesis, lambda is a function of QP[9]:

$$\lambda = 0.85 \times 2 \times e^{(QP-12)/3}, \ \ where \ 0 \le QP \le 51$$

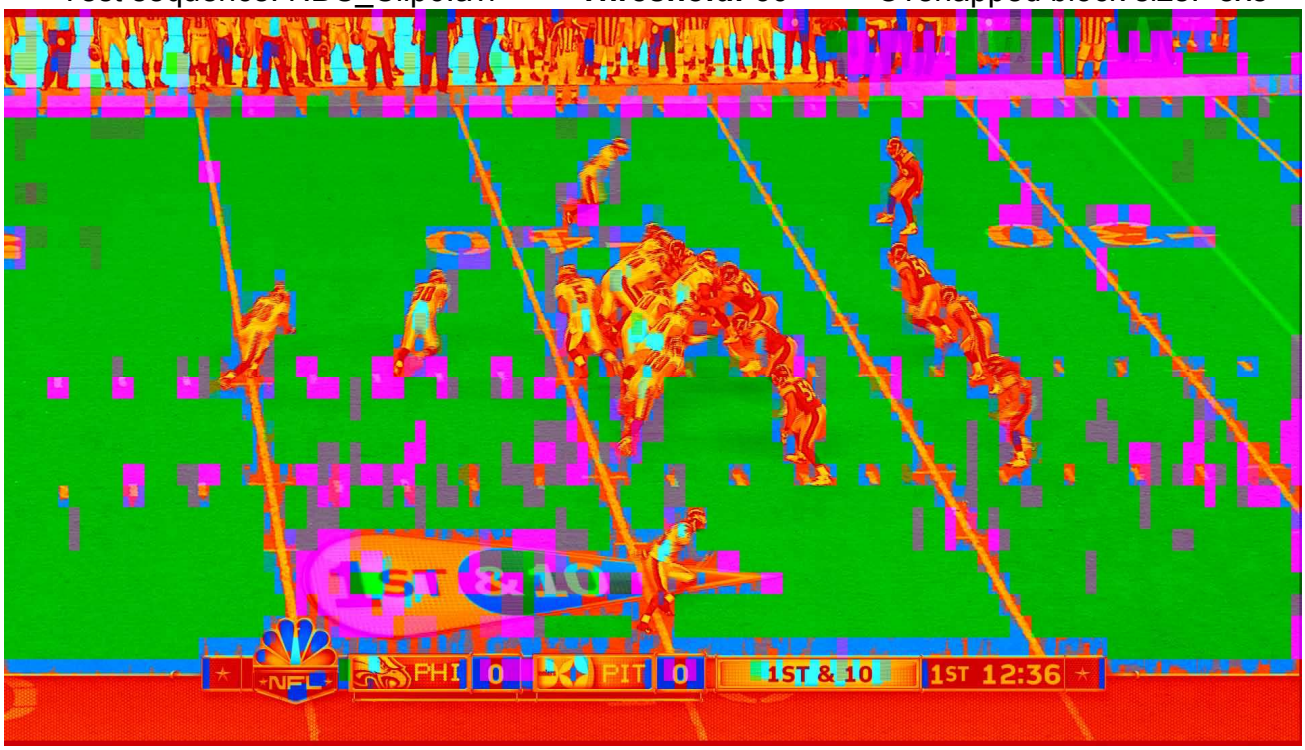
QP	**0**	**1**	**2**	**3**	**4**	**5**	**6**
λ	0.053125	0.066933	0.084331	0.106250	0.133867	0.168661	0.212500
QP	**7**	**8**	**9**	**10**	**11**	**12**	**13**
λ	0.267733	0.337323	0.425000	0.535466	0.674645	0.850000	1.070933
QP	**14**	**15**	**16**	**17**	**18**	**19**	**20**
λ	1.349291	1.700000	2.141866	2.698582	3.400000	4.283732	5.397164
QP	**21**	**22**	**23**	**24**	**25**	**26**	**27**
λ	6.800000	8.567463	10.794327	13.600000	17.134926	21.588654	27.200000
QP	**28**	**29**	**30**	**31**	**32**	**33**	**34**
λ	34.269853	43.177309	54.400000	68.539705	86.354617	108.80000	137.079410
QP	**35**	**36**	**37**	**38**	**39**	**40**	**41**
λ	172.709234	217.60000	274.158820	345.418469	435.20000	548.317641	690.836938
QP	**42**	**43**	**44**	**45**	**46**	**47**	**48**
λ	870.400000	1096.635282	1381.673876	1740.80000	2193.270564	2763.347751	3481.60000
QP	**49**	**50**	**51**				
λ	4386.541127	5526.695503	6963.20000				

Appendix B: The Optimal Threshold Test

To find the optimal threshold for the distortion compensation algorithm, several values are tested (Threshold = 55, 60, 65, 70 or 75). Test results are included here.

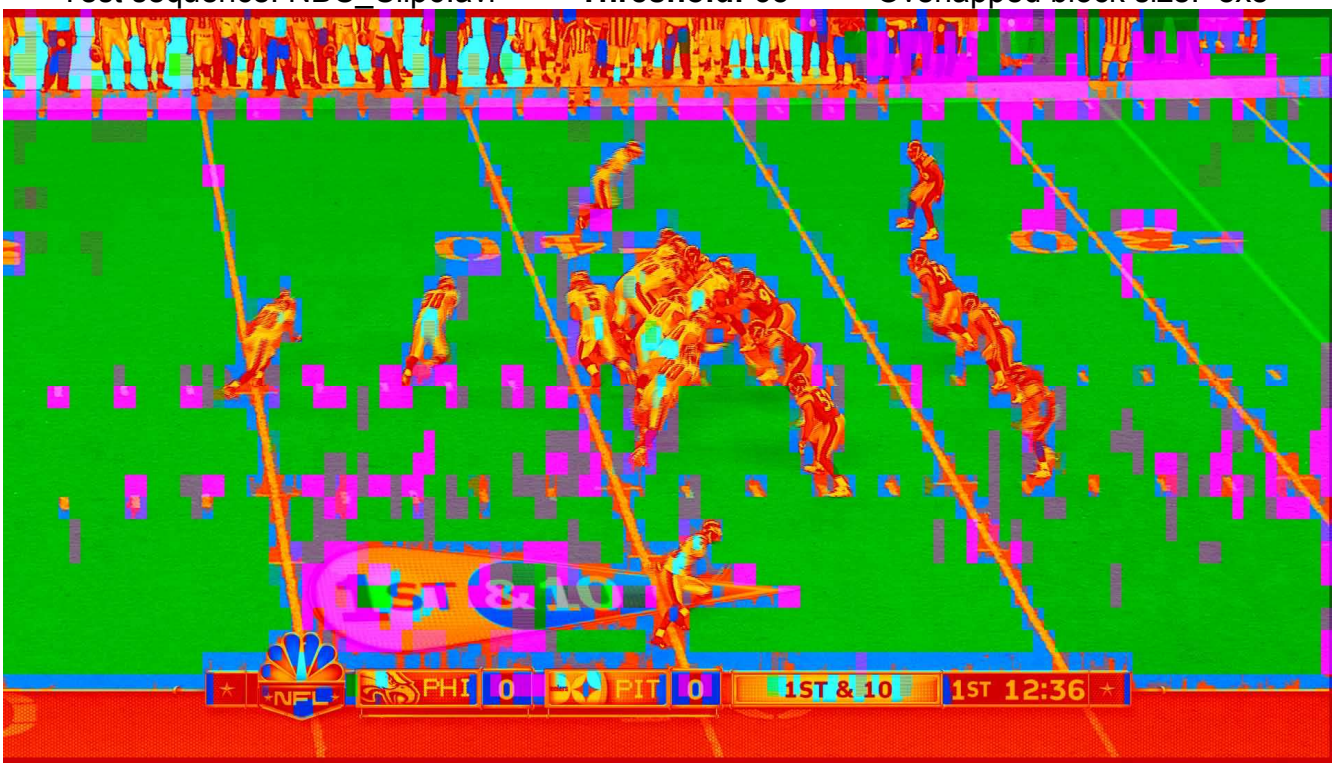
- **Test E-1: Find the Optimal Threshold**
- Test sequence: NBC_Clip6.avi **Threshold: 55** Overlapped block size: 8x8

- **Test E-2: Find the Optimal Threshold**
- Test sequence: NBC_Clip6.avi **Threshold: 60** Overlapped block size: 8x8

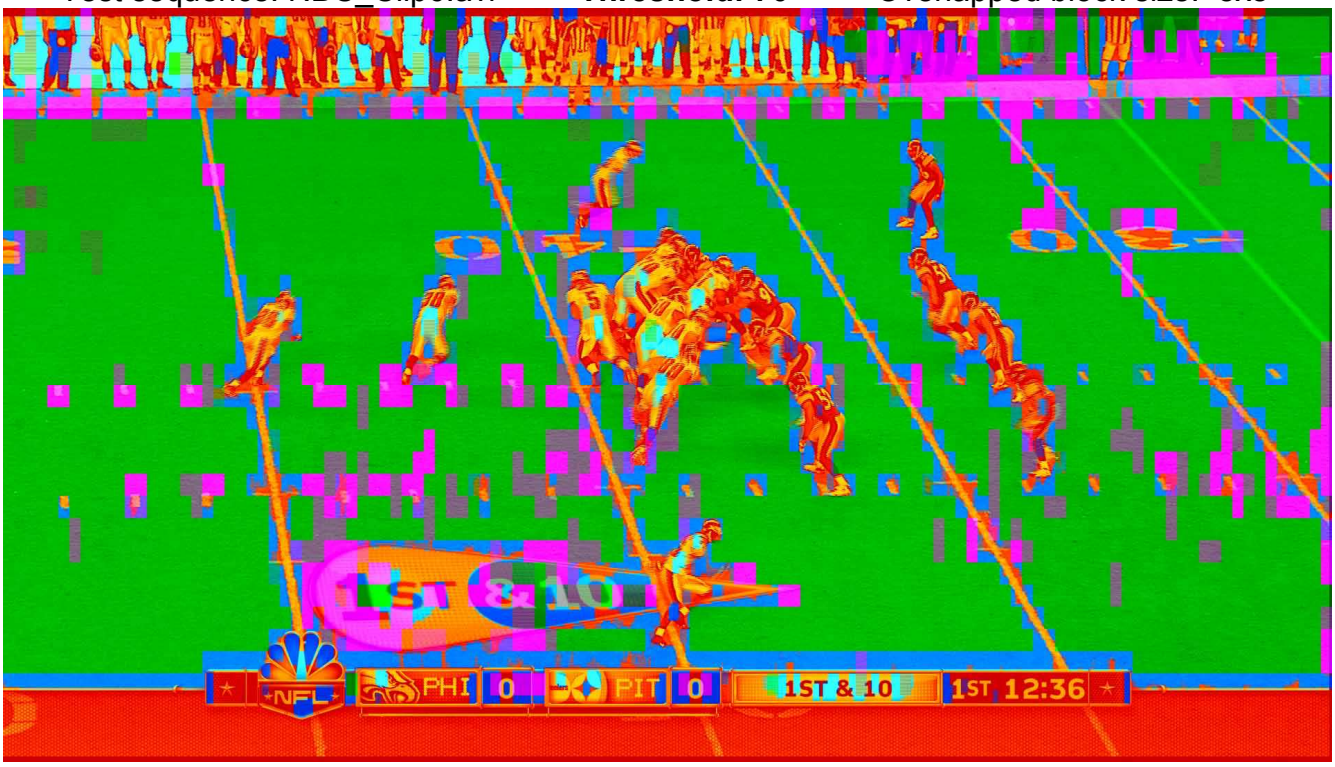


Test E-3: Find the Optimal Threshold

- Test sequence: NBC_Clip6.avi **Threshold: 65** Overlapped block size: 8x8



- **Test E-4: Find the Optimal Threshold**
- Test sequence: NBC_Clip6.avi **Threshold: 70** Overlapped block size: 8x8



- **Test E-5: Find the Optimal Threshold**
- Test sequence: NBC_Clip6.avi **Threshold: 75** Overlapped block size: 8x8