UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

André Beims Bräscher

Evaluation of Motion Estimation Lagrange Multipliers in HEVC

Florianópolis
2020

André Beims Bräscher

Evaluation of Motion Estimation Lagrange Multipliers in HEVC

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Ciência da Computação.
Orientador: Prof. José Luís Almada Güntzel, Dr.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Bräscher, André Beims
Evaluation of motion estimation Lagrange Multipliers in HEVC / André Beims Bräscher ; orientador, José Luís Almada Güntzel, 2020.
58 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2020.

Inclui referências.

1. Ciência da Computação. 2. Codificação de Vídeos. 3. HEVC. 4. Multiplicadores de Lagrange. 5. SATD. I. Güntzel, José Luís Almada. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

André Beims Bräscher

Evaluation of Motion Estimation Lagrange Multipliers in HEVC

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Alex Sandro Roschildt Pinto, Dr.
Universidade Federal de Santa Catarina

Prof. Guilherme Ribeiro Corrêa, Dr.
Universidade Federal de Pelotas

Prof. Mateus Grellert da Silva, Dr.
Universidade Católica de Pelotas

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Ciência da Computação.

Documento assinado digitalmente
ANTONIO AUGUSTO MEDEIROS FROHLICH
Data: 14/01/2020 14:04:29-0300
CPF: 500.696.400-68

Prof. Antônio Augusto Medeiros Fröhlich, Dr.
Coordenador do Programa

Documento assinado digitalmente
Jose Luis Almada Guntzel
Data: 14/01/2020 13:43:52-0300
CPF: 533.780.670-04

Prof. José Luís Almada Güntzel, Dr.
Orientador

Florianópolis, 2020.

ACKNOWLEDGEMENTS

This work was supported by the Research and Innovation Support Foundation of the State of Santa Catarina (FAPESC). This study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

RESUMO

Vídeos digitais demandam bastante espaço para armazenamento e largura de banda para transmissão, o que pode ser tratado codificando os dados de maneira a comprimi-los. A solução do estado da arte para codificação de vídeos é conhecida como *High Efficiency Video Coding* (HEVC). O HEVC incorpora várias ferramentas que o tornam altamente adaptável a diferentes tipos de conteúdo de vídeo. Porém, tais ferramentas requerem a tomada de diversas decisões durante o processo de codificação. Por exemplo, o *HEVC Model* (HM), que é o *software* de referência do HEVC, toma tais decisões considerando os impactos tanto em distorção quanto em taxa de *bits* através do algoritmo de Otimização de Taxa-Distorção (RDO). No RDO, um Multiplicador de Lagrange (λ) é aplicado sobre os valores de taxa a fim de definir a importância (peso) da taxa de *bits* em relação à distorção. Todavia, ainda há a necessidade de definir um λ de forma a dar um peso adequado à taxa de *bits* em relação à distorção. Pode-se encontrar diversos trabalhos da literatura fornecendo um embasamento de como definir valores de λ considerando duas das métricas usadas no HM, Soma das Diferenças Quadráticas (SSD) e Soma das Diferenças Absolutas (SAD). Porém, na literatura há uma lacuna de trabalhos avaliando valores para λ considerando a métrica de Soma das Diferenças Absolutas Transformadas (SATD). Tal lacuna na literatura é especialmente significativa pois a SATD é a métrica padrão de cálculo de distorção para o RDO na etapa de Estimação de Movimentos Fracionária (FME) do HEVC. Neste trabalho, foram realizados dois conjuntos de experimentos com fatores multiplicativos constantes (m) aplicados sobre os valores de λ padrões a fim de avaliar os impactos em eficiência do HM ao mudar λ. O primeiro conjunto (Cenário-I) consistiu em multiplicar por m os valores de λ usados na ME, tanto com a SAD quanto com a SATD. No segundo conjunto apenas os valores de λ usados na etapa de FME (portanto, usando a SATD) foram multiplicados por m no (Cenário-II). No caso do Cenário-I, percebeu-se que multiplicadores com valores na faixa $0.9 \leq m \leq 1.2$ produziram resultados similares (incluindo a referência com $m = 1$). Por outro lado, o Cenário-II apresentou ganhos em eficiência de codificação do HM ao aumentar os valores de λ na FME por um fator de $m = 1.3$. Nós também amostramos a distorção e a taxa a partir da função de FME durante a execução dos testes descritos. A partir dos dados amostrados, nós buscamos correlações entre distorção ou taxa com o melhor valor de m para cada vídeo testado. Porém, nenhuma das métricas avaliadas apresentou correlação com os melhores valores de m, contradizendo assim as afirmações feitas em um dos trabalhos correlatos mais relevantes.

Palavras-chave: HEVC 1. Estimação de Movimento 2. Otimização de Taxa Distorção 3. Multiplicadores de Lagrange 4. SAD 5. SATD 6.

RESUMO EXPANDIDO

Introdução

Vídeos digitais demandam bastante espaço para armazenamento e largura de banda para transmissão, o que pode ser tratado codificando os dados de maneira a comprimi-los. A solução do estado da arte para codificação de vídeos é conhecida como *High Efficiency Video Coding* (HEVC). Tal padrão de codificação usa uma arquitetura conhecida como arquitetura híbrida de codificação, a qual é composta por quatro etapas básicas: predição, transformada, quantização e codificação de entropia. A primeira etapa busca, dentre um conjunto de blocos candidatos, aquele que resulta em melhor substituição do candidato original. Assim, o candidato escolhido passa a ser denominado "bloco de referência". A substituição de um bloco original é feita usando a diferença entre tal bloco e o respectivo bloco de referência (a diferença é chamada de resíduo), juntamente com informações necessárias para a obtenção do bloco de referência. A etapa de transformada tem o papel de transformar os resíduos do domínio espacial para o domínio de frequências, melhorando a eficácia das etapas subsequentes. A terceira etapa (quantização) aplica divisões sobre as amostras de acordo com um Parâmetro de Quantização (QP), o que introduz erros pois tais divisões possuem precisão finita. Porém, a quantização potencialmente acarreta dois efeitos importantes: (I) redução do número de bits necessários para representar um resíduo e (II) redução do número de símbolos diferentes representando os resíduos a serem codificados. A última etapa é a codificação de entropia, a qual opera sobre todos os dados a serem codificados (ao contrário das etapas de estimação e transformada). Esta etapa final reduz o número de bits resultantes de codificações ao substituir os símbolos mais comuns por códigos que usam menos bits enquanto atribui códigos com mais bits para representar símbolos menos usados. Além disso, a etapa de predição é dividida em dois modos, intra e inter, sendo que uma decisão de modo posteriormente escolhe entre a referência da intra e da inter. No caso da predição intra, os candidatos são gerados a partir do mesmo quadro do bloco original. A predição inter, por sua vez, obtém os candidatos a partir de quadros previamente codificados e é subdividida em duas etapas: IME e FME. Considerando todas essas etapas, muitas decisões que afetam a eficiência de codificação precisam ser tomadas durante o processo de codificação. Uma solução clásica para este problema consiste em considerar os impactos tanto em distorção (D) quanto em taxa de *bits* (R) através do algoritmo de Otimização de Taxa-Distorção (RDO), de acordo com um custo $J = D + \lambda \times R$. No RDO, um Multiplicador de Lagrange (λ) é aplicado sobre os valores de taxa a fim de definir a importância (peso) da taxa de *bits* em relação à distorção. Todavia, ainda há a necessidade de definir como os valores compondo o custo J são obtidos. No caso do *HEVC Model* (HM), que é o *software* de referência do HEVC, os valores de distorção e taxa são obtidos diferentemente, conforme a etapa de predição. Desta forma, também são necessários diferentes valores para λ de forma a manter relações adequadas de peso entre a taxa de *bits* e a distorção. Por isso, neste trabalho é adotada uma notação para representar a métrica de distorção e a etapa em que um determinado Multiplicador de Lagrange é usado. No caso da decisão de modo usando a métrica chamada de Soma das Diferenças Quadradas (SSD), a notação usada é $\lambda_{mode,SSD}$. Durante a IME, por sua vez, a decisão usa a métrica de Soma das Diferenças Absolutas (SAD), e assim, a notação usada é $\lambda_{motion,SAD}$. No caso da FME, a métrica utilizada é a Soma das Diferenças Transformadas Absolutas (SATD) e por isso, a notação usada é $\lambda_{motion,SATD}$. Pode-se

encontrar diversos trabalhos da literatura fornecendo um embasamento de como definir valores para $\lambda_{mode,SSD}$ e $\lambda_{motion,SAD}$. Porém, há uma lacuna de trabalhos avaliando valores para $\lambda_{motion,SATD}$. Considerando os problemas mencionados na literatura a respeito de ajustes de Multiplicadores de Lagrange, há a necessidade de mais informação nos impactos de tais valores. Portanto, este trabalho busca clarificar os impactos de se variar o peso da componente *R* no cálculo de custo do RDO. Além disso, foi realizada uma análise aprofundada do caso de usar a métrica de SATD, por se tratar da maior lacuna na literatura. Por isso, este trabalho tem como escopo os valores usados na predição inter, $\lambda_{motion,SAD}$ e $\lambda_{motion,SATD}$, especialmente com respeito ao Multiplicador de Lagrange a ser usado em conjunto com a SATD.

Objetivos

O objetivo geral deste trabalho é analisar as repercuções, em termos de eficiência de codificação, de se variar os Multiplicadores de Lagrange a serem usados com o RDO, considerando a SATD como métrica de distorção. A fim de atingir o objetivo geral, foram realizadas análises mais específicas avaliando os seguintes aspectos: (I) a equação para λ considerando a SATD de maneira análoga à definição da equação para o caso da SSD; (II) quais valores de λ resultam em melhor eficiência de codificação; (III) correlações entre características como o conteúdo do vídeo, a taxa ou a distorção e os valores de λ que resultam em melhor eficiência.

Metodologia

Primeiramente, foram consideradas tanto abordagens analíticas quanto empíricas. Um dos principais trabalhos analíticos foi apresentado por Gish e Pierce (1968) e propôs um modelo de distorção em função de quantização, sendo genérico quanto à medida de erro. T. Wiegand e Girod (2001) adaptaram o modelo de Gish e Pierce (1968) para o caso da SSD e, com base nisso, desenvolveram um modelo para $\lambda_{mode,SSD}$ em função do quadrado da quantização. Além disso, T. Wiegand e Girod (2001) também propuseram um ajuste para $\lambda_{motion,SAD}$ usando a raiz quadrada de $\lambda_{mode,SSD}$. Tal ajuste pode ser justificado pela diferença do cálculo entre as métricas de SSD e SAD aonde a primeira faz o quadrado das diferenças, enquanto que a segunda calcula o absoluto. Por conta dos trabalhos de Gish e Pierce (1968) e T. Wiegand e Girod (2001) parece natural a possibilidade de adaptar tais formulações para o caso da SATD. Porém, não há um ajuste trivial, como a aplicação de uma raiz quadrada, para modelar a etapa extra de transformada. Além disso, pode-se considerar adaptar a SATD no modelo de T. Wiegand e Girod (2001) calculando uma integral pixel a pixel sobre os erros, conforme o modelo de Gish e Pierce (1968). Esta segunda alternativa analítica tem por problema o fato de que a transformada da SATD é aplicada aos erros como um todo e gera uma dependência espacial entre todos os pixels de um bloco sendo processado. Por isso, não foi possível calcular a integral da modelagem original e optou-se por buscar uma solução empírica. Desta forma, é preciso definir os experimentos a serem realizados para a obtenção de dados empíricos. Neste trabalho, foram realizados dois conjuntos de experimentos com fatores multiplicativos constantes (*m*) aplicados sobre os valores de λ padrões a fim de avaliar os impactos em eficiência do HM (versão 16.15) ao mudar λ. O primeiro conjunto (Cenário-I) consistiu em multiplicar por *m* os valores de λ usados na predição inter, ou seja, tanto $\lambda_{motion,SAD}$ quanto $\lambda_{motion,SATD}$. No segundo conjunto (Cenário-II), apenas os valores de λ usados na etapa de FME (portanto, $\lambda_{motion,SATD}$) foram multiplicados por *m*. Para cada cenário foram executados testes com as mesmas

configurações, considerando um conjunto de 22 valores para m e 4 QPs para codificar 47 sequências de vídeo. Desta forma, cada cenário foi avaliado considerando 4136 execuções do HM, totalizando 8272 execuções ao todo.

Resultados e Discussão

No caso do Cenário-I, percebeu-se que os multiplicadores no intervalo $0.9 \leq m \leq 1.2$ produziram resultados similares (incluindo a referência com $m = 1$). Por outro lado, o Cenário-II apresentou ganhos em eficiência de codificação do HM ao aumentar os valores de λ na FME por um fator de $m = 1.3$. Durante a execução dos testes, a distorção e a taxa foram amostradas a partir da função de FME do HM. A partir dos dados amostrados, foram buscadas correlações entre distorção ou taxa com o melhor valor de m para cada vídeo testado. Porém, nenhuma das métricas avaliadas apresentou correlação com os melhores valores de m, contradizendo assim as afirmações feitas em um dos trabalhos correlatos mais relevantes.

Considerações Finais

Este trabalho proporcionou um melhor entendimento a respeito do comportamento da eficiência de compressão no HM com diferentes Multiplicadores de Lagrange, além de indicar uma possível melhoria ao aumentar o multiplicador por um fator de $m = 1.3$. Além disso, pode-se identificar pelo menos três possibilidades de trabalhos futuros a partir deste trabalho. Primeiramente, pode-se realizar testes similares com resoluções mais altas para avaliar se as conclusões obtidas a partir de vídeos de baixa resolução se confirmam em contextos de alta resolução. Outra possibilidade de trabalho futuro está em avaliar possíveis interações entre diferentes combinações de multiplicadores para os λs usados com a SAD e a com a SATD na ME. Finalmente, existe a possibilidade de que, ao combinar mais variáveis, obtenha-se informação suficiente para encontrar correlações entre as variáveis e os melhores multiplicadores de Lagrange.

Palavras-chave: HEVC 1. Estimação de Movimento 2. Otimização de Taxa Distorção 3. Multiplicadores de Lagrange 4. SAD 5. SATD 6.

ABSTRACT

Digital videos demand large storage space and high bandwidth to be transmitted, which can be addressed by coding the data in a compressing manner. The state-of-the-art solution to video coding is the standard dubbed High Efficiency Video Coding (HEVC). HEVC incorporates various tools making it very adaptable to different kinds of video content but also requiring several choices to be made during the coding process. For instance, the HEVC Model (HM), which is HEVC's reference software, makes such choices by considering the impacts to both distortion and bit rate through what is known as Rate-Distortion Optimization (RDO). In RDO, a Lagrange Multiplier (λ) is applied to the rate values in order to define the importance (weight) of the bit rate in the trade off with the distortion. Nevertheless, the necessity remains for defining λ such that it weighs adequately the relationship between rate and distortion. We can find in the literature a number of works providing some background on how to define the values of λ when considering the Sum of Squared Differences (SSD) and Sum of Absolute Differences (SAD) distortion metrics, two of the used metrics in HM. However, there is a lack of works evaluating λ values considering the Sum of Absolute Transformed Differences (SATD). Such a gap in the literature is especially significant because the SATD is the default distortion measure for the RDO in the HEVC step known as Fractional Motion Estimation (FME). In this work we conducted two sets of experiments with constant multiplicative factors (m) applied to the default values of λ in order to evaluate the impacts of changing λ in the performance of HM. The first set (Scenario-I) consisted in multiplying the λs employed with both SAD and SATD in ME by m. Meanwhile, in Scenario-II only the λs used in the FME step (i.e., with the SATD) are multiplied by m. The obtained results show that λs between 0.9 and 1.2 produce similar results (including the baseline with $m = 1$) under Scenario-I. On the other hand, Scenario-II demonstrated coding efficiency gains on HM when increasing λ by a factor of 1.3 for the FME computation. During the execution of the described tests we also sampled distortion and rate data from the FME computation function. With the sampled data we searched for correlations between distortion or rate to the best m value for each tested video sequence. However, none of the tested metrics showed correlation to the best ms, which contradicts the claims made by one of the most relevant related works.

Keywords: HEVC 1. Motion Estimation 2. Rate-Distortion Optimization 3. Lagrange Multipliers 4. SAD 5. SATD 6.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AVC	Advanced Video Coding
BD-Rate	Bjøntegaard Delta Bitrate
CALM	Context Adaptive Lagrange Multiplier
CTC	Common Test Conditions
FME	Fractional Motion Estimation
fps	frames per second
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
HM	HEVC Model.
IME	Integer Motion Estimation
ISO	International Standardization Organization
ITU	International Telecommunication Union.
JM	Joint Model
LD	Low-delay
MATE	Mean Absolute Transformed Error
ME	Motion Estimation
MSE	Mean Squared Error
PSNR	Peak signal-to-noise ratio
QP	Quantization Parameter
RA	Random-access
RDO	Rate-Distortion Optimization
SAD	Sum of Absolute Differences
SATD	Sum of Absolute Transformed Differences
SSD	Sum of Squared Differences
TI	Temporal Information

CONTENTS

1 INTRODUCTION

Videos in their raw state (i.e., devoid of any compression) demand massive storage and/or bandwidth. Equation 1 exemplifies the bit rate for raw Full-HD videos at 30 frames per second (fps) and 24 bits per pixel.

$$\begin{aligned} \text{Bit rate} &= \overbrace{\text{Height} \times \text{Width}}^{\text{Resolution}} \times \text{Bits per Pixel} \times \text{fps} \\ &= 1080 \times 1920 \times 24 \times 30 \\ &\approx 1,5\text{Gbits/s} \end{aligned} \tag{1}$$

Such situation can be alleviated by coding the video deploying a set of tools able to compress the amount of data needed to represent it. By compressing a video without distorting the image (i.e., lossless coding), the coding efficiency is improved. This idea can be extended by allowing lossy coding and accounting for bit rate as well as the amount of distortion inserted. The concept of coding efficiency allows for comparing the results between different compression solutions (e.g. comparing implementations of a coding standard).

Currently, the state-of-the-art video coding standard is the so-called High Efficiency Video Coding (HEVC) (SULLIVAN, G. J.; OHM, et al., 2012). Such standard was developed by a joint team with experts from International Standardization Organization (ISO) and International Telecommunication Union (ITU) aiming at improving coding efficiency by 50% compared to its predecessor, the Advanced Video Coding (AVC)[1] (SULLIVAN, Gary J., 2005). Rather than defining an implementation, the HEVC standard only specifies the bit stream (i.e., the way information has to be coded). Consequently, there is a degree of flexibility to the development of video coders for such standard. Another characteristic of HEVC is its high adaptability to both bit rate constraints and video content. Accordingly, compliant coders can be very adaptable to coding requirements and video characteristics, which is exemplified by HEVC's reference coder, HEVC Model (HM)[2] (JCT-VC, 2013). Meanwhile, the reference coder for AVC is known as Joint Model (JM) (JVT, 2011).

The way HM adapts itself to different situations is by making several coding decisions at execution time, while still producing video codings in accordance with the HEVC standard. However, there is the matter of how to make good decisions at execution time. This problem can be tackled by using the idea of coding efficiency in the form of Rate-Distortion Optimization (RDO). The classical formulation for RDO is as a constrained problem, presented in the following equation (SULLIVAN, G. J.; WIEGAND, T., 1998):

$$min\{D\}, \text{ subject to } R < R_c \tag{2}$$

[1] The AVC and HEVC standards are also known as H.264 and H.265, respectively.

[2] Henceforth, this work uses HM as the base video coder.

where, D and R correspond to, respectively, the calculated distortion[3] and bit rate, while R_C is a rate constraint.

Formulating RDO as a constrained problem has some limitations and may not be very practical. An alternative is to incorporate the constraint from Equation 2 in the minimization using a Lagrange Multiplier (λ) as in the following equation (SULLIVAN, G. J.; WIEGAND, T., 1998):

$$min\{J\}, \text{ where } J = D + \lambda \times R \tag{3}$$

therefore, in this formulation J is a cost to be minimized. Section 2.2 further discusses particularities about the RDO and how it fits in different steps of the coding process.

With the latter formulation, each λ value is guaranteed to provide the optimal solution for an unknown rate constraint. Still, it is necessary to decide which λ value should be used in a given situation.

The HM implementation of Equation 3 optimizes constants to be adopted in various scenarios which will be discussed in Section 3.2. Unfortunately, to the best of our knowledge, such optimization process has not been detailed in the literature. This problem persists for more specific adjustments as, for instance, when the **Sum of Absolute Transformed Differences (SATD)** is to be used as distortion metric (subsection 2.3.4). In such case, the HM documentation defines an adjustment factor of 0.95 multiplying the corresponding values employed for the Sum of Absolute Differences (SAD) (subsection 2.3.1) distortion metric, once again with scarce explanation as to why.

Such problems also occur in some of the classical works that defined the basics of λ adjustment. For instance, T. Wiegand and Girod (2001) claimed that, when using SAD as distortion metric, the values should be the square root of those obtained for the Sum of Squared Differences (SSD) (subsection 2.3.2) distortion metric. However, the authors' claim was based on unrevealed empirical data, thus proposing a certain adjustment without disclosing what led to that.

Fortunately, there are works providing solid reasoning behind their proposals for λ computation. However, they may arrive into opposing conclusions. Such was the case with (ZHANG, J. et al., 2010) and (ZHANG, F.; BULL, 2018). The former claimed λ values should be increased with larger motion, whereas the latter defended that they should be smaller with dynamic content. Furthermore, another issue pertains to the values to be used in conjunction with the SATD. Apart from the aforementioned adjustment applied by HM, there is limited discussion on this matter.

Considering the mentioned issues with the literature regarding Lagrange Multiplier adjustments, there is a need for more information on the impacts of such values. Therefore, we aim to shed more light into the impacts of varying the weight over the R

[3] Section 2.3 presents a discussion on a few distortion metrics.

term from Equation 3. Furthermore, we went into more details on the case of using the SATD metric, since the largest gap in this area surrounds such scenario.

1.1 GOALS

1.1.1 Main Goal

This work's main goal is to analyze the coding efficiency repercussions from varying the Lagrange multipliers to be used in RDO when SATD is employed as distortion metric.

1.1.2 Specific Goals

Aiming to achieve the main goal, we have to perform more specific analyses by evaluating the following aspects:

- The Equation for λ considering the SATD in an analoguos way as defined for the SSD;
- Which λ values result in better coding efficiency;
- Correlations between video content, rate, or distortion characteristics with the λ values that result in better efficiency.

1.2 CONTRIBUTIONS

The analyses performed in this work improve the understanding of RDO in the context of Motion Estimation (ME), especially considering the Fractional Motion Estimation (FME) using SATD. Moreover, the results showed that the coding efficiency can be improved by changing the Lagrange multiplier employed with the SATD. Finally, by better understanding the problem of λ optimization, we identified inconsistencies in the state-of-the-art literature.

1.3 ORGANIZATION

The remainder of this work is organized by, firstly, elaborating on the essential concepts in Chapter 2. Subsequently, Chapter 3 expands on the brief overview of the works regarding λ optimization while presenting a rough flow of λ formulations over the years. Next, Chapter 4 details the method employed for our experimentations allowing the analyses featured in Chapter 5. Finally, the conclusions we arrived at are drawn in Chapter 6.

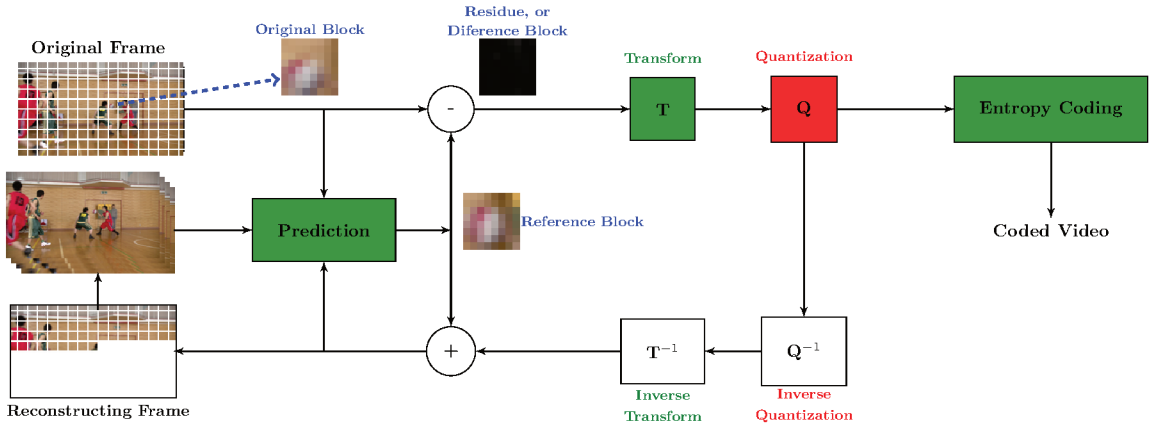
2 BASIC CONCEPTS

This chapter is aimed at providing the basic concepts necessary to better understand the environment to which this work pertains. For a deeper discussion on the basic concepts of video coding we suggest the textbook (RICHARDSON, 2002). Also, for more details specifically to the HEVC we point out the textbook from Sze, Budagavi, and Gary J Sullivan (2014). The remainder of this chapter is organized starting with a discussion of the hybrid coding architecture used by modern video coders in Section 2.1. Afterwards, Section 2.2 further elaborates on RDO and how it works in HM. Then, Section 2.3 presents definitions for the distortion metrics mentioned in Chapter 1. Section 2.4 discusses the different types of frames considered through the coding process. Finally, Section 2.5 features an overview of common coding configurations employed for the HM.

2.1 HYBRID VIDEO CODING

Both, AVC and HEVC coders follow the hybrid coding architecture, incorporating prediction and transform techniques. As can be seen in Figure 1, such architecture is comprised of four basic steps: prediction, transform, quantization and entropy coding. The first step searches among a set of candidate blocks, the one providing the best substitution to an original block to become a reference block. The candidate blocks may be obtained directly from previously coded pixels or through predetermined modes of generating blocks (mostly through interpolations) Such substitution is performed by using the difference between the reference and original blocks (residue) along with the information necessary to obtain the reference block. The transform step has the role of transforming the residues from the spatial domain to the frequency domain, improving the efficacy of the subsequent steps. The third step (quantization) applies divisions according to a Quantization Parameter (QP) so as to reduce the magnitudes of the transformed residues. Because the divisions in the quantization step have finite precision, such step ends up by introducing errors to the encoded video. However, by performing such divisions the quantization potentially leads to two important features: (I) reduction of number of bits necessary to represent a given residue and (II) reduction of number of different symbols representing the residues to be coded. The final step is the entropy coding, e.g., arithmetic coding (RICHARDSON, 2002). Entropy coding operates on all the data to be encoded, unlike the transform and the quantization steps, which are applied only to the residues. This final step reduces the overall number of bits for the resulting coding by substituting the more commonly present symbols by codes that use few bits and choosing codes that use more bits to represent symbols that are rarely used.

Figure 1 – The hybrid coding model.



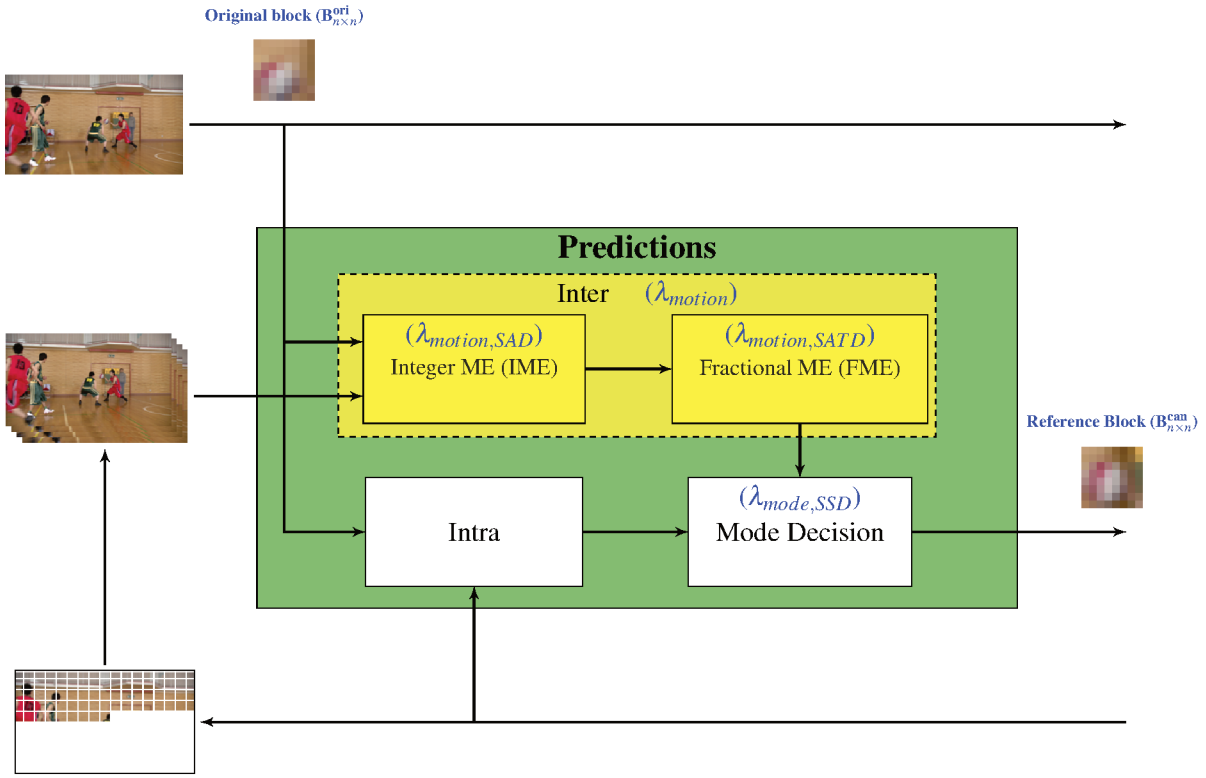
Source: adapted from Richardson (2003).

2.1.1 Prediction

Elaborating on the prediction step, it is divided in two modes according to how the candidates are obtained. The intra prediction aims at reducing spatial redundancies through candidates each generated with a different mode of intra prediction. Whereas, the inter prediction's goal is to purge temporal redundancies through the process known as Motion Estimation (ME). Because of that, inter prediction relies on data from other frames to generate the candidates. Still, the ME can also be further divided into Integer Motion Estimation (IME) and Fractional Motion Estimation (FME). The former consists on obtaining the candidates from blocks of previously coded frames and the latter further refines it with new candidate blocks generated through interpolation of pixels. Such interpolation produces the effect of fractionally positioned pixels, possibly around the best candidate from IME. The analyses featured in this work were made within the context of FME. Figure 2 depicts the organization of the prediction with yellow highlighting on the scope of this work.

The prediction step requires many decisions for each original block with several candidates from intra and even more during inter prediction. Additionally, there is the choice of which prediction mode (e.g. intra vs. inter modes) to be used which is known as mode decision. An intuitive way of making all of those decisions in the prediction step is to consider their impact on the coding efficiency using Equation 3. In the following section, we present the RDO approaches adopted in HM at the mode decision and inter estimation steps.

Figure 2 – Prediction modes.



Source: adapted from Seidel (2015)

2.2 RATE DISTORTION OPTIMIZATION

This section addresses matters of implementation of the RDO specifically to the HM[1]. HM implements RDO as an unconstrained problem as defined in Equation 3. However, there is still the matter of choosing which cost function to be used at different coding steps. Such choice has to take into account how accurate are the measures of rate and distortion impacts as well as the coding time required. In the remainder of this section, we are going to limit our discussion to the RDO employed for the mode decision and for the inter prediction steps since the scope of this work is limited to the latter step.

For the mode decision, HM calculates the distortion using the SSD metric (Subsection 2.3.2) applied on reconstructed candidate blocks. A block is said to be reconstructed after being computed through the transform, quantization, inverse quantization and inverse transform steps, effectively coding and decoding a block. Reconstructing a block allows for better distortion estimation by accounting for the losses through the coding process. Similarly, the resulting rate for a candidate is obtained after transforming, quantizing and applying entropy encoding considering the candidate block and coding structures necessary to represent it (e.g. motion vector). By adopting such approaches

[1] Since the RDO is only applied to guide coding decisions it only affects the coding process, thus, it is not part of the HEVC standard.

to the computation of rate and distortion, the RDO is based on accurate measures for its components. The drawback of performing all those steps for computing rate and distortion lies on the significant computational effort required. Because the mode decision considers fewer candidates this approach still is feasible.

Unfortunately, it is not viable to adopt the same level of thoroughness during inter prediction due to the many candidates that have to be evaluated, thus, requiring a few simplifications. Firstly, both the distortion and rate are estimated without applying the subsequent coding steps to the candidates. In the case of the distortion, it is computed directly between the candidate and original blocks using the SATD (Subsection 2.3.4, for FME) or even the SAD (Subsection 2.3.1, for IME) metrics, both of which are easier to compute. As for the rate, it is estimated from the size of the motion vector for the candidate being tested. Such simplifications greatly reduce the computational effort per candidate at the expense of worse estimations on the coding efficiency implications for the evaluated choices.

Because of the differences to the way the components from Equation 3 are calculated the λs also have to be different. Therefore, henceforth we address the specific λ for mode decision as λ_{mode} while using λ_{motion} for the inter prediction. Furthermore, because the HM adopts different metrics for IME and FME their corresponding λs are going to be referred to as $\lambda_{motion,SAD}$ and $\lambda_{motion,SATD}$, respectively. The various λ designations for the implementations of Equation 3 at the corresponding coding steps are also depicted in Figure 2.

2.3 DISTORTION AND AVERAGE ERROR METRICS

This section is aimed at providing a brief overview of the distortion metrics employed in HM for RDO computation (i.e. SAD, SSD and SATD), as discussed in the previous section. Furthermore, we also present two metrics for average error per pixel, the Mean Squared Error (MSE) and the Mean Absolute Transformed Error (MATE), which are going to be used in Chapter 5.

2.3.1 Sum of Absolute Differences (SAD)

SAD is a very simple metric. The first step (shown in Equation 4) is to calculate the differences between the pixels of an original block ($\mathbf{B}^{\text{ori}}_{N\times O}$) and a candidate block ($\mathbf{B}^{\text{can}}_{N\times O}$). Meanwhile, henceforth, N and O respectively represent the number of rows and columns in pixel matrices being processed.

$$\mathbf{B}^{\text{diff}}_{N\times O} = \mathbf{B}^{\text{can}}_{N\times O} - \mathbf{B}^{\text{ori}}_{N\times O} \tag{4}$$

Afterwards, the absolute of each difference is computed and added up according

to the following equation:

$$sad = \sum_{i=1}^{N} \sum_{j=1}^{O} |d_{i,j}| \tag{5}$$

, with $d_{i,j}$ being the element in the i-th row and j-th column from a $\mathbf{B}^{\text{diff}}_{N \times O}$ differences matrix.

2.3.2 Sum of Squared Differences (SSD)

The SSD is quite similar to the SAD except that the differences are squared instead of taken the absolute. With that, small homogeneous changes in the blocks are favored over big discrepancies and, therefore, the perceptual quality[2] tends to be improved. However, the downside is that square operations demand considerably more computational resources than absolute operations. The equation for the SSD computation is as follows:

$$ssd = \sum_{i=1}^{N} \sum_{j=1}^{O} d_{i,j}^2 \tag{6}$$

2.3.3 Mean Squared Error (MSE)

The MSE corresponds to the average squared difference per pixel and, therefore, corresponds to the SSD for a pair of blocks divided by the number of pixels per block, as shown in Equation 7.

$$mse = \frac{\sum_{i=1}^{N} \sum_{j=1}^{O} d_{i,j}^2}{N \times O} = \frac{ssd}{N \times O} \tag{7}$$

2.3.4 Sum of Absolute Transformed Differences (SATD)

Equation 8 shows the SATD computation. The SATD works very similarly to the SAD, but with an extra transform step before the absolute computation. This metric is expected to provide better quality than the SAD (AKRAMULLAH, 2014). A possible explanation for the improved quality of the SATD over the SAD is that it may improve the correlation to the coded video by approximating the transform step from the coding process with its own transform.

$$satd = c \sum_{i=1}^{N} \sum_{j=1}^{N} |td_{i,j}| \tag{8}$$

[2] Here, perceptual quality concerns the quality perceived by the human visual system.

, where c is a scaling factor (according to the size of the transform matrix) and the elements $td_{i,j}$ belong to the matrix obtained according to the following equation:

$$\mathbf{TD}_{N\times N} = \mathbf{T}_{N\times N} \times \mathbf{B}^{\text{diff}}_{N\times N} \times \mathbf{T}^{\mathsf{T}}_{N\times N} \tag{9}$$

, with T being a transformation matrix. Using the HM as example, it adopts the Hadamard matrix as transform for SATD computation. Such matrix can be defined according to various orders for $2^k \times 2^k$ sizes with $k \in \mathbb{N}^+$. Equation 10 exemplifies with a 4×4 Hadamard.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{10}$$

2.3.5 Mean Absolute Transformed Error (MATE)

Similarly to the relation between the MSE and SSD, it can be drawn the MATE[3] as the SATD for a pair of blocks divided by the number of pixels per block, as shown in Equation 11.

$$mate = \frac{c\sum_{i=1}^{N}\sum_{j=1}^{N}|td_{i,j}|}{N\times N} = \frac{satd}{N\times N} \tag{11}$$

2.4 FRAME TYPES

The HEVC defines three frame types (I, P and B) with a type being assigned for each frame in a video sequence according to the coding configurations (Section 2.5).

I-frames are the most elementary ones because they predetermine the prediction mode to only intra-frame. In fact, primitive digital video coding was performed by processing frames as if they were a sequence of unrelated pictures. At first, using I-frames seem to be a limitation to coding efficiency optimization, yet this type of frame is necessary for a couple of reasons. The first and main reason is for coding the first frame. In fact, the coding execution is constrained to intra prediction because there is no other frame to be referenced in inter prediction. Therefore, such situations require the use of I-frames. Another reason for using I-frames is to improve robustness when decoding. Such reason would not be a factor in an ideal world where no frames are ever lost and the decoding always starts at the beginning of the coded stream. However, by adding I-frames the length of data dependency chains is capped by the interval between I-frames, hence limiting reconstruction errors due to data losses only to the interval containing such losses.

[3] The MATE metric is not widely adopted (in fact, to the best of our knowledge, this metric has not been previously defined in the literature). However, it was employed in this work to better evaluate distortion results across blocks of different sizes in Chapter 5.

The P-frames allow both intra and inter predictions with the latter having a single list of reference frames for the reconstruction of inter-predicted blocks. By allowing the choice of inter-prediction during the coding process, these frames can provide significant coding efficiency gains. The drawbacks from using this kind of frame are reduced robustness and more coding computational effort due to the extra prediction mode to be processed as well as the subsequent mode decision for each block.

B-frames are very similar to P-frames but use two lists of references which potentially improves coding efficiency at the cost of more computational effort from considering more references. Additionally, F. Zhang and Bull (2018) classified B-frames as B_p when a B-frame references only temporally preceding frames and B_b for instances referencing both temporally preceding and succeeding frames. Though, such classification was only used to aid in the development and implementation of their method and did not affect the HM behavior in itself.

2.5 CODING CONFIGURATIONS

The configurations for the HM define several important parameters for each frame to be encoded, including the types of frames, QP offsets and frame hierarchic levels, which are encompassed in Group of Pictures (GOP) structures. This section is going to provide a background on such parameters exemplifying with the Common Test Conditions (CTC) (BOSSEN, 2012) configurations for the HM. The CTC configurations are divided into three groups: (I) Intra, (II) Low-delay (LD) and (III) Random-access (RA). Regarding the CTC, they propose a set of configurations and video sequences to be used with HM testing, which facilitates the commonality between works with such software.

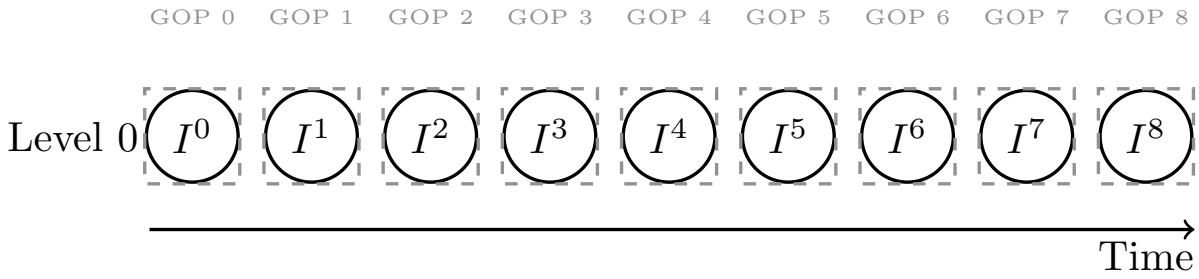
2.5.1 Intra

With the intra configuration, as its name suggests, all frames are of I type and each frame is contained in its own single-frame-sized GOP. Such characteristics also eliminate the possibility of having QP offsets or different frame levels. Figure 3 shows a diagram of a sequence of nine frames using the intra configuration, where each vertex indicates the coding order along the video sequence time axis, while the dashed rectangles indicate the GOPs.

2.5.2 Low-delay

The Low-delay (LD) configuration consists in having the mandatory initial intra frame and all of the remaining frames as either P-frames (Low-delay P) or B-frames (Low-delay B). With this configuration, the frames can only reference preceding frames in display order and the two reference lists on B-frames are identical. Therefore, using

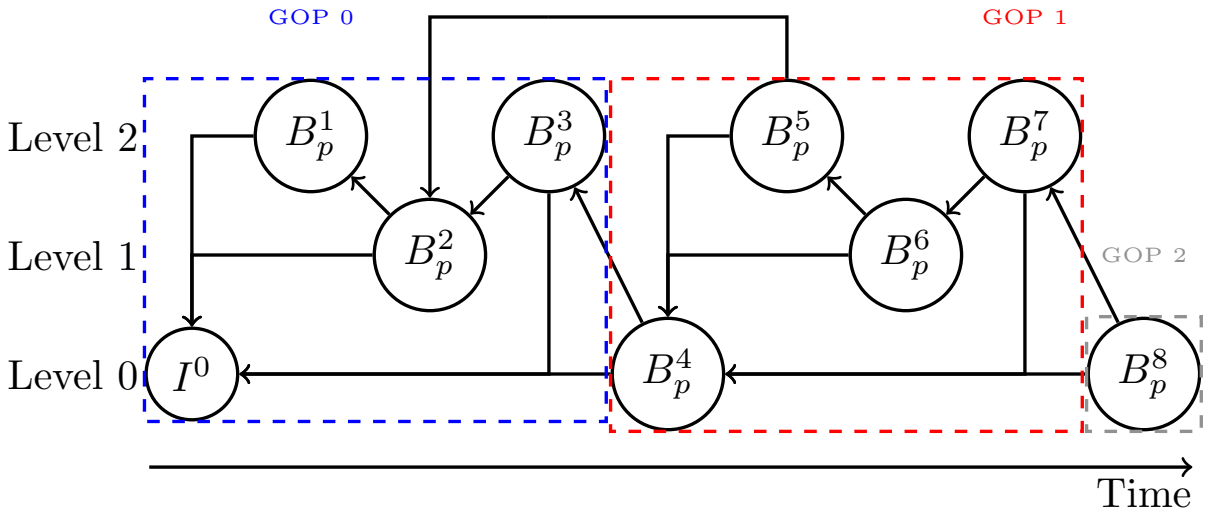
Figure 3 – Intra configuration example.



Source: the author.

the classification from F. Zhang and Bull (2018) all B-frames in Low-delay B end up being B_p frames. Furthermore, frames are distributed into three levels according to the number of referencing frames pointing to them. Therefore, frames contained in higher levels are less oftenly reference by other frames and, because of the reduced importance to the overall coding efficiency, they are coded with higher QP offsets. Figure 4 shows a diagram considering nine frames to be encoded divided in three GOPs using LD configuration based on the example in (JCT-VC, 2013). In such figure, the vertices indicate the frames in coding order along the time axis. Furthermore, each vertex contains the type of frame it represents as well as its position in coding order. Meanwhile, the edges indicate the references (from referencing frames pointing to frames being referenced). Finally, the dashed rectangles indicate the frames contained in a same GOP.

Figure 4 – Example of Low-delay (LD) configuration using B-frames.

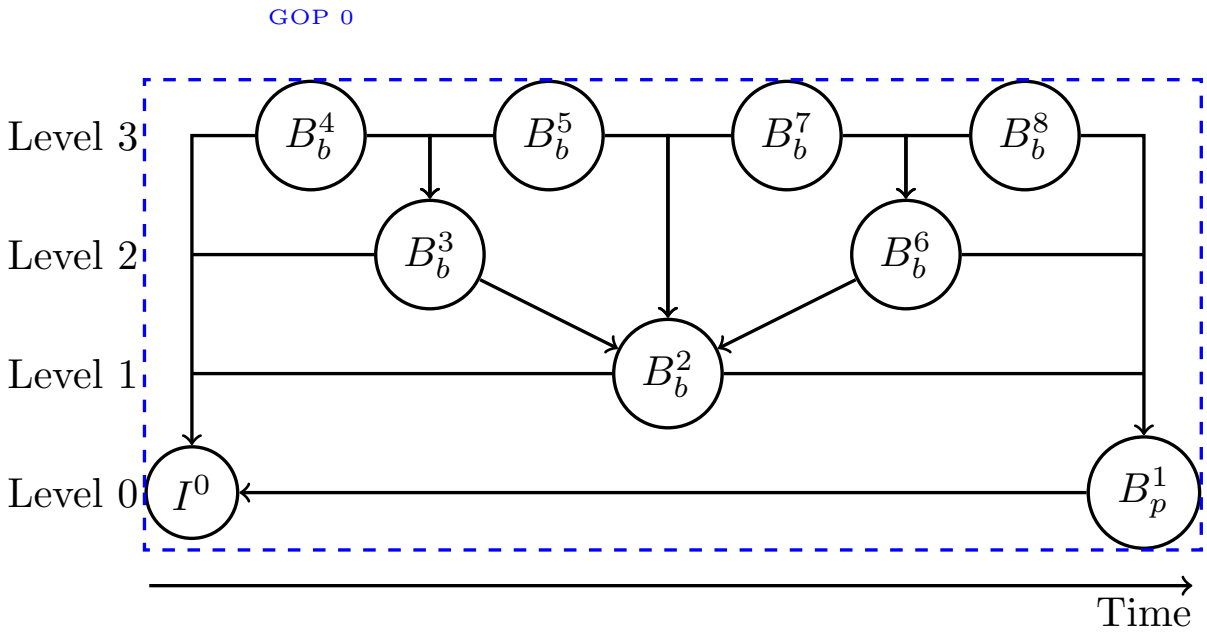


Source: the author. Based on the example from (JCT-VC, 2013).

2.5.3 Random-access

The Random-access (RA) configuration is centered around a hierarchical structure with B-frames and I-frames. Such hierarchy allows for better coverage of temporally near frames as reference frames, including temporally succeeding frames. Furthermore, because the information coded on higher hierarchical frames are referenced by fewer frames (up to the point where they are not referenced at all on the highest hierarchical level), they use higher QPs through QP offsets. In the case of the CTC configuration employed in this work, the GOP size is 16 frames with intra period of 32 frames (i.e. one I-frame at every 32 frames). Since a full RA GOP would require representing a structure with 16 frames, Figure 5 presents a simplified version with 9 frames based on the example in (JCT-VC, 2013). Similarly to Figure 4, the vertices in Figure 5 represent the frames along the sequence time, with each vertex containing the frame type and coding order position of the frame it represents. Furthermore, the edges in Figure 5 point from referencing frames to frames being referenced.

Figure 5 – Example of Random-access (RA) configuration with a simplified GOP.



Source: the author. Based on the example from (JCT-VC, 2013).

3 REFERENCE SOFTWARE AND RELATED WORKS

This chapter presents a few of the main works regarding RDO using Lagrange multipliers. Firstly, in Section 3.1, we present the early works developing the basis for state-of-the-art models. Afterwards, we discuss the values adopted by the JM and by the HM in Section 3.2. Then, Section 3.3 concerns proposals for adjustments to the λ choices in the reference software. Finally, Section 3.4 covers the development of alternative models to the ones in Section 3.2.

3.1 EARLY WORKS

G. J. Sullivan and T. Wiegand (1998) provided an in-depth view of the theory behind RDO and the arguments for using a Lagrange multiplier (λ) formulation. Such work analyzed the application of RDO at different parts of the coding process. The authors also addressed the determination of a cost function to be used in RDO that uses SSD as distortion metric. Said cost function determines the relation of importance between rate and SSD distortion with λ defined as a function of the quantization level. Finally, they proposed applying a square root over the default SSD λ to account for the metric change when using the SAD.

T. Wiegand and Girod (2001) further detailed how to get to similar formulations as the ones proposed by G. J. Sullivan and T. Wiegand (1998). The presented logic is based on the idea that the Lagrange multiplier corresponds to the slope of the rate-distortion function. Then, they showed that if the distortion function $D(R)$ is strictly convex, then $J_{mode}(R) = D(R) + \lambda_{mode} \times R$ is strictly convex, with the *mode* designation indicating that the cost function is being used for mode decision. From that, assuming $D(R)$ to be differentiable and equating the cost to 0 as to get the function minimum, the following equation was obtained:

$$\lambda_{mode} = -\frac{\mathrm{d}D}{\mathrm{d}R} \tag{12}$$

Still, remained the need for determining D and R functions with the latter being defined as a function of the former in the following equation:

$$R(D) = a \log_2 \left(\frac{b}{D} \right) \tag{13}$$

where, a and b are parameters which should be determined.

As for the distortion function, it was used the model from Gish and Pierce (1968). In such work, the authors presented a generic method considering an error measure L (e.g. squared difference) that satisfies the following restrictions:

1. $L(0) = 0$;

2. L is an increasing function of the magnitude of its argument;

3. The function $M(v) = \left({}^{1}/_{v}\right) \int_{-v/2}^{v/2} L(u)\,\mathrm{d}u$ satisfies: $xM'(x)$ is monotone.

The error measure is a way of calculating distortions to be accumulated in an M function. It can be seen that the described M is an average distortion function (e.g. MSE, MATE) rather than a sum of all distortions (e.g. SSD, SATD). In fact, the authors presented the process of obtaining the distortion function by using the MSE as an example. An issue with the presented method is that Gish and Pierce (1968) considered their own assumptions for the density function of the variable being quantized to be "rather weak".

When T. Wiegand and Girod (2001) adapted M from the MSE to the SSD, they arrived at:

$$D = \frac{(2Q)^2}{12} = \frac{Q^2}{3} \tag{14}$$

, with Q corresponding to the quantization step size and its relationship to the QP is defined in Equation 16.

Then, by getting the derivatives for D and R and substituting them into $\mathrm{d}D$ and $\mathrm{d}R$ from Equation 12, the following equation was obtained:

$$\lambda_{mode}(Q) = -\frac{\mathrm{d}D(Q)}{\mathrm{d}R(Q)} = \frac{\ln 2}{3a} \times Q^2 = c \times Q^2 \tag{15}$$

Still, there is the need to find out what is the best value for c. It is important to notice that the constant c in Equation 15 has nothing to do with the scaling factor used in SATD (Equation 8) and, henceforth, c is going to represent the constant in the λ function. According to T. Wiegand and Girod (2001), the best experimental results use 0.85, though they did not present the process used to obtain such value.

In summary, for the model derived by T. Wiegand and Girod (2001), the following assumptions were made:

1. Distortion function $D(R)$ is strictly convex and differentiable;

2. Uniform distribution within each quantization interval.

It is important to notice that the use of such assumptions was criticized by Syu (2005) as discussed in Section 3.4.

In the context of AVC and HEVC, the quantization is configured through the QP which has the following relation to Q (SZE; BUDAGAVI; SULLIVAN, Gary J, 2014):

$$Q = 2^{\frac{(QP-4)}{6}} \tag{16}$$

Because of that, the original formulation can be adapted to λ as a function of QP as specified in Equation 17.

$$\lambda_{mode}(QP) = c \times 2^{\frac{(QP-12)}{3}} \tag{17}$$

3.2 REFERENCE SOFTWARE

Equation 17 shows how the base λ function is computed during FME on a P-frame in JM[1], according to (LIM; SULLIVAN, G.; WIEGAND, Thomas, 2006). JM's roots in the work from T. Wiegand and Girod (2001) are clear in Equation 18, including the use of a square root for λ_{motion} to compensate for the fact that the SAD does not have the square operation from the SSD.

$$\lambda_{motion,P}(QP) = \sqrt{\underbrace{0.85}_{c} \times \underbrace{2^{\frac{(QP-12)}{3}}}_{Q^2}} \tag{18}$$

A further multiplication is then made for FME computation on B-frames (LIM; SULLIVAN, G.; WIEGAND, Thomas, 2006):

$$\lambda_{motion,B}(QP) = \sqrt{max\left(2, min\left(4, \frac{(QP-12)}{6}\right)\right) \times \lambda_{mode,I,P}} \tag{19}$$

The *min* and *max* functions are effectively limiting the left hand multiplication factor to the range [2,4]. Thus, it only varies the multiplication in the QP interval [24,36], while clipping the other cases. Such adjustment means that the priority for the rate component of the cost (Equation 3) is increased two to four times.

Revisiting λ functions employed by HM, it bases the calculation on the following equation:

$$\lambda_{mode}(QP) = \underbrace{\alpha \times W_k}_{c} \times \underbrace{2^{\frac{(QP-12)}{3}}}_{Q^2} \tag{20}$$

With α being (MCCANN et al., 2014):

$$\alpha = \begin{cases} 1 - Clip3(0, 0.5, 0.5 \times \#B\_frames) & \text{for referenced pictures} \\ 1 & \text{for non-referenced pictures} \end{cases} \tag{21}$$

While α assumes the multiplication identity for non-referenced frames, it can vary from 1 to 0.5 within referenced pictures. Table 1 presents how the values for W_k are employed.

[1] As it was commented in Chapter 1, the JM is the reference software for the AVC standard, analogously to the HM for the HEVC.

Table 1 – W_k values. *Clip* corresponds to the $Clip3\left(2,4,\frac{(QP-12)}{6}\right)$ operation. Adapted from (MCCANN et al., 2014).

k	QP offset hierarchy level	Slice Type	Referenced	W_k	W range
0	0	I	–	0.57	0.57
1	0	P or B	1	RA: 0.442	0.442
				LD: 0.578	0.578
2	1, 2	P or B	1	RA: $0.3536 \times Clip$	[0.7072, 1.4144]
				LD: $0.4624 \times Clip$	[0.9248, 1.8496]
4	3	B	0	RA: $0.68 \times Clip$	[1.36, 2.72]

There is a clear influence from HM's predecessor (JM) through the common clippings between Table 1 (when $k \geq 2$) and Equation 19. In HM, such clipping actually changes the function from Equation 15 since the c component becomes proportional to QP instead of being a constant. Furthermore, both JM and HM **adopted the proposed square-root adjustment when using SAD for distortion estimation**. However, despite the similarities, HM clearly is more complex in regards to adjusting λ to different scenarios. Such adjustments take into account the frame level within its respective GOP, the number of B frames and the general coding configuration (either RA or LD).

In HM, the range of values to form the c portion of Equation 20 is quite significant. By computing such terms, it can be seen that λ_{mode} may be as low as 0.2221 with α clipped to 0.5 and using $W_1 = 0.442$ at RA configuration. With similar reasoning, c may be as high as 2.72 with W_4 and $QP \geq 36$ (so that $Clip = 4$). Besides having a higher amplitude, the magnitude of the values is also quite different to JM, with the latter having c in the interval between [1.7, 3.4].

Still, on top of all that, **HM employs an adjustment when using the SATD** in relation to the default λ used with SAD prediction. The λs to be used according to distortion metric are shown in Table 2 (MCCANN et al., 2014).

Table 2 – HM's λ values according to distortion metric.

SSD	SAD	SATD
λ_{mode}	$\lambda_{motion,SAD} = \sqrt{\lambda_{mode}}$	$\lambda_{motion,SATD} = \sqrt{0.95 \times \lambda_{mode}}$

There are a couple of reasons for all the differences between the adopted λs for JM and HM. The first simply being that those are different coding standards and coding process distinctions impact the rate-distortion results. Furthermore, there has clearly been an effort to optimize the λ to be adopted by HM in different scenarios, whereas JM only took the QP and type of frame into account.

3.3 ADJUSTING THE λ MODEL

This section discusses the works from J. Zhang et al. (2010), González de Suso Molinero (2016) and F. Zhang and Bull (2018). Such works proposed adjustments to reference software's λ values and, while (ZHANG, J. et al., 2010) was based on JM, the other two performed analyses with both JM and HM.

3.3.1 CALM Method

J. Zhang et al. (2010) started by performing two rounds of tests. The first round consisted on multiplying the default $\lambda_{motion,SAD}$ values by a constant belonging to $\{0.5, 0.8, 1, 1.5, 2\}$. Such tests were performed for only two frames from 3 sequences with the first frame being an I frame and the second a P-frame. The second round of experiments used a similar setup but coding 30 frames and varying QP values. As a result, the authors made the following statements:

1. The original λ function produces near optimum rate-distortion curves when the motions are small across a frame or sequence.
2. When the motion vectors either have a more random behavior or are larger, increasing λ leads to a better rate-distortion curve.

Having those two statements as guidelines, the authors developed what was dubbed the Context Adaptive Lagrange Multiplier (CALM) method. The aim was to dynamically adjust λ values at the macroblock level. For that, the authors defined a threshold where, while the cost estimate is below such threshold ($J_{threshold}$)[2], no change is made. However, when the cost estimate is above the threshold, a multiplicative factor (F_{CALM}) defined in Equation 22 is applied on the default λ.

$$F_{CALM} = \sqrt{\frac{J_{neighborhood}}{J_{threshold} \times (0.3 \times QP - 6)}} \tag{22}$$

Where $J_{neighborhood}$ is derived from the cost estimates from left and above neighboring macroblocks. If none of those blocks are available, $J_{neighborhood} = J_{threshold}$. The CALM method was tested in the context of ME and using the default estimators for distortion (i.e. SAD) and rate. Then, the authors performed a final round of tests to compare results with and without the proposed method. Such round considered 12 sequences at 30 fps with up to 100 frames and QPs in the set $\{24, 28, 32, 36\}$. The authors reported results within [–0.64%, 0.05%] Bjøntegaard Delta Bitrate (BD-Rate) and averaging –0.25% BD-Rate (i.e., reduced bit rate compared to the baseline), both on the luma channel.

[2] The authors suggested using $J_{threshold} = 512$.

The results obtained showed slight improvements in coding efficiency based on neighboring prediction data. Furthermore, by using the cost from Equation 3 they accounted for both distortion and rate changes. However, such approach may lead to efficiency losses over time because of the feedback loop from previously set λs affecting newer λs through the cost equation. Moreover, an important limitation from (ZHANG, J. et al., 2010) is the lack of test data. In such work, the authors relied on at most 30 frames from each of 3 sequences amounting to a total of 90 frames tested before making the guiding statements.

3.3.2 Background Adaptive and Multiple Estimation Methods

González de Suso Molinero (2016) performed analyses with constant multiplying factors on both JM and HM. For each software they applied separately the factors to the default λ_{mode}s and $\lambda_{motion,SAD}$s, amounting to four separate sets of tests. Each set of experiments considered four QP values and five multiplying factors between 0.5 and 2.1 and six sequences at CIF (352×288) resolution.

Concerning JM, the author found out that the bigger potential for improvements was to change the $\lambda_{motion,SAD}$. He also proposed performing the ME with three λ conditions: (I) the default values, (II) distortion based cost ($\lambda_{motion,SAD} = 0$) and (III) rate based cost ($\lambda_{motion,SAD} \rightarrow \infty$). The choice between which λ (i.e. candidate) to be used at each case was left to the mode decision, which means using the more complex RDO from such step to finalize the motion decision. The proposed method achieved 2.2% reduction (improvement) to the BD-Rate while increasing the coding time by approximately 3%. The same experiment setup for evaluation of his own proposal was also used to evaluate the CALM method through which the author found out that the CALM method did not improve the coding efficiency while increasing the coding time by 0.24%.

In the case of the HM testing, an issue is that the author did not address the matter of the SATD and the *HadamardME* employed by default in HM's FME. Therefore it is unclear if the experiments affected both the $\lambda_{motion,SAD}$ and $\lambda_{motion,SATD}$ or only the former. Regardless of that, the author concluded that the bigger potential for improvements lied in changing the λ_{mode}. Furthermore, the author claimed that there is a correlation between the results varying the λ_{mode} and the presence of static backgrounds in the video content being coded. Because of that, González de Suso Molinero (2016) proposed a background adaptive method. In such method, he implemented a detection of static backgrounds and when such type of background was identified the coder used an adapted λ through a regression function used to determine a multiplier. In practice, the proposed method increases the λ values when coding more static scenes, giving higher importance to the rate estimates. The proposed method resulted in lower coding efficiency with the sequences classified as containing dynamic background but

higher efficiency in the cases classified as containing static backgrounds. Furthermore, the author showed that when using higher values for λ_{mode}, the HM execution time was reduced. Therefore, the proposed method also reduced coding time. Finally, the author compared his proposal to the one in (ZHAO et al., 2013) and showed that his proposal resulted in better coding efficiency.

3.3.3 Scene Adaptive Method

The adopted scope in the work of F. Zhang and Bull (2018), was the $\lambda_{motion,SAD}$ from Table 2. They started by comparing rate-distortion figures from various tested λs (λ_{test}) to the original ones (λ_{orig}) so that $0.2 \leq \frac{\lambda_{test}}{\lambda_{orig}} \leq 5$. This preliminary test involved 9 sequences at CIF (352×288) resolution, with 4:2:0 YUV chroma sampling, QPs in $\{27, 32, 37, 42\}$ and QP offset disabled[3]. The authors also classified the video sequences in three groups according to how dynamic the videos are. Furthermore, they performed the experiments with five different configurations to evaluate the results for different GOP structures and I, P and B-frames. Additionally, they separated B-frames into B_p (predicting from temporally past frames) and B_b (predicting from temporally past and future frames).

Considering the test results, the authors were able to select λ values yielding the best overall rate-distortion performance for all frames, though such selection process was not disclosed. Nevertheless, the authors drew the following conclusions:

1. The original λ values perform well when coding I, P and B_p frames.
2. The original λs are inadequate when coding B_b frames, particularly for either static or highly dynamic videos (contrarily to the mixed content class). The former kind of video requires higher values while the opposite yields better results in the latter situation.

Therefore, F. Zhang and Bull (2018) opted to develop a method to address the second observation. Further investigating the causes for bad performance with B_b frames, they defined the following ratios:

$$r_{MSE} = \frac{MSE_p}{MSE_B} \tag{23}$$

$$r_{\lambda} = \frac{\lambda_{opt}}{\lambda_{orig}} \tag{24}$$

[3] By disabling the QP offset, all frames in a given coder execution use the same QP, instead of varying QP according to each frame's hierarchy in its GOP.

Where MSE_p corresponds to the MSE of P and B_p frames, while MSE_b corresponds to the MSE of B_b frames. Both λs in Equation 24 regarded B_b frames. Then, they used the r_{MSE} to fit a function to the best results from previous tests:

$$r_{\lambda,fit} = f(r_{MSE}) = a(r_{MSE} + d)^b + c \tag{25}$$

With a, b, c, d being constants dependent on the coding software (either JM or HM) and GOP size. Moreover, because r_{MSE} is dependent on previously coded data, the $r_{\lambda,fit}$ is updated with each coded frame. The application of such constants is exemplified in Equation 26, considering HM execution with a GOP size of eight.

$$r_{\lambda,fit} = 2.197(r_{MSE} + 0.04)^{5.196} + 0.308 \tag{26}$$

F. Zhang and Bull (2018) proposed an algorithm using the model from Equation 25. Such algorithm assumes that, provided that there is no scene cut, the characteristics between neighboring frames remain similar, including the previously defined ratios. Because of that, the authors used a scene cut detection logic to choose between using the standard λ and one obtained from Equation 25. Aside from scene cuts, the standard values are also employed while coding non B_b frames as well as when there is not enough previous frame information. Finally, the results confirmed the effectiveness for the proposed approach by achieving average BD-Rate reductions between 1% and 1.2% with the HM using the CTC and RA configuration.

Despite the work from F. Zhang and Bull (2018) clearly achieving improvements in coding efficiency, it is important to point out to a few gray areas in their work. Firstly, the way by which the best results were picked from the initial tests and then used as basis for the proposed model. Reportedly, the work considered the optimality within each tested QP, which rules out the use of BD-Rate. Therefore, by not using a metric that accounts for both rate and distortion they had to directly use rate and distortion data. Consequently, it became a multi-objective optimization problem. However, the paper does not discuss such issue and thus, it is not clear how it was addressed.

Another issue concerns the claim that r_{MSE} is able to detect how dynamic a scene is. However, such ratio is based on a distortion metric which is tied to how good a prediction is. Therefore, a relatively highly dynamic scene may still be coded adequately and result in relatively low distortions. Furthermore, changing λ shifts the importance between distortion and rate, hence, affecting the distortion values. Consequently, using such ratio as basis for λ computation may result in changes trickling down to frames yet to be coded.

Finally, there is also the question of diverging conclusions between the work from J. Zhang et al. (2010) and F. Zhang and Bull (2018). While the former proposed increasing λ values when the vectors are random or large (which can be caused by highly

dynamic contexts), the latter concluded that λ should be reduced in highly dynamic scenes.

3.4 CHANGING THE λ MODEL

Sangi, Heikkila, and Silven (2004) set out to provide better λ choices by changing the underlying model altogether. They created an alternative to Equation 12 through linear approximation and a few simplifications to the problem. Regarding the distortion metric, both SSD and SAD were taken into account. The proposed method performed similarly in comparison to the standard model, with the authors claiming that the results depended on the content of the sequence. However, despite such claim, they evaluated the performance with only one test sequence (Foreman). Furthermore, the authors stated that a wide range of λ values can be safely used in practice due to the inconclusiveness of their tests. Still, there was no analysis of the λ values coverage throughout the tested approaches. Therefore, it is not possible to draw such conclusions about the coding efficiency across a wide range of λ values.

Syu (2005) developed new functions to model the distortion and rate estimates when using the SAD as well as the λ in a similar way to (WIEGAND, T.; GIROD, 2001) in Equation 3. However, the authors experimentally identified issues both with the SAD correlation to the distortion as well as with the λ function from T. Wiegand and Girod (2001). For the distortion function it was claimed that there are “unkwnown” factors (besides the SAD and QP) affecting the results. Regarding the function for λ, it was observed that a few of the assumptions made in (WIEGAND, T.; GIROD, 2001) do not hold. Furthermore, the authors attempted to obtain a better λ function based on their own rate and distortion functions. Unfortunately, new issues arrived as having to obtain the correct values (or functions) for parameters in the derived functions. Finally, they ran a simulation for their new λ function with a few approximations. The authors did not disclose the results from such simulations, however they considered the results to be unsatisfactory.

Deng et al. (2013) proposed an alternative rate-distortion function. Their proposal was to have a base model and to define a set of function parameters with pre-coding data. Such pre-coding meant coding frames with up-to 5 QP values while gathering the rate, SSD and SATD distortion measures. This approach achieved average BD-Rate reduction (improved coding efficiency) by 2.59% however it also doubled the already long coding time from the default HM.

3.5 LITERATURE LIMITATIONS

The main limitation in the literature for λ values determination lies on the lack of works addressing the λs to be used with the SATD metric. Furthermore, a few

methodological issues were noticed during the exploration of the works aiming at better values for the cost equation with SSD and SAD, including the limited number of video sequences considered. Finally, such works have arrived at seemingly contradictory conclusions reinforcing the need for further exploration on the matter.

Table 3 summarizes the discussed related works. The first column shows either which kind of λ in the reference software was addressed or (in the analytical works) for which distortion metric were the alternative functions proposed. The column "Approaches" characterizes the works by the adopted approach wherein analytical approaches are the ones proposing changes to λ by means of mathematical analyses whereas empirical approaches propose alterations based on experiments collecting data from software executions. In the latter cases, it is also important to know how the experiments were set up. Therefore, we summarized the experiments by the numbers of constant multiplying factors, sequences and QPs. Finally, various works proposed adaptive adjustments suggesting that m should be either increased or decreased according to how dynamic was the video content. In Table 3, we represent with ↓ a relatively low quantity (i.e., lower movements or lower m values), meanwhile, ↑ represents relatively higher quantities and finally, = is used to represent the use of unchanged λ values (i.e., $m = 1$).

Table 3 – Related works summary.

	Scope	Approaches	Reference software		Empirical testing (#)			Adaptive adjustment	
			JM	HM	m	sequences	QPs	movement	m
SANGI; HEIKKILA; SILVEN (2004)	SSD | SAD	Analytical	✓	—	—	—	—	—	—
SYU (2005)	SAD	Analytical	✓	—	—	—	—	—	—
DENG et al. (2013)	SSD | SATD	Analytical	—	✓	—	—	—	—	—
ZHANG, J. et al. (2010)	$\lambda_{motion,SAD}$	Empirical	✓	—	5	3	4	↓ ↑	= ↑
GONZÁLEZ DE SUSO MOLINERO (2016)	$\lambda_{mode,SSD}$ | $\lambda_{motion,SAD}$	Empirical	✓	✓	5	6	4	↓ ↑	↑ =
ZHANG, F.; BULL (2018)	$\lambda_{motion,SAD}$	Empirical	✓	✓	?	9	4	↓ ↑	↑ ↓

Firstly, we brought up a broad combination of scopes and approaches and the only case that is not featured in the previous table is an empirical analysis of $\lambda_{motion,SATD}$ which, to the best of our knowledge, has not been done before. Furthermore, there are a few works that only considered the JM and, thus, might change in the context of HM, especially in the empirically approached case (ZHANG, J. et al., 2010). Regarding the empirical testing, all of the discussed works lack in the number of experiment cases. Firstly, such works considered few constant multipliers, therefore, painting a vague picture of the coding efficiency over different λ values. The number of considered video sequences was low as well with the most being nine and even getting to the extreme of just three sequences being considered to draw correlations between

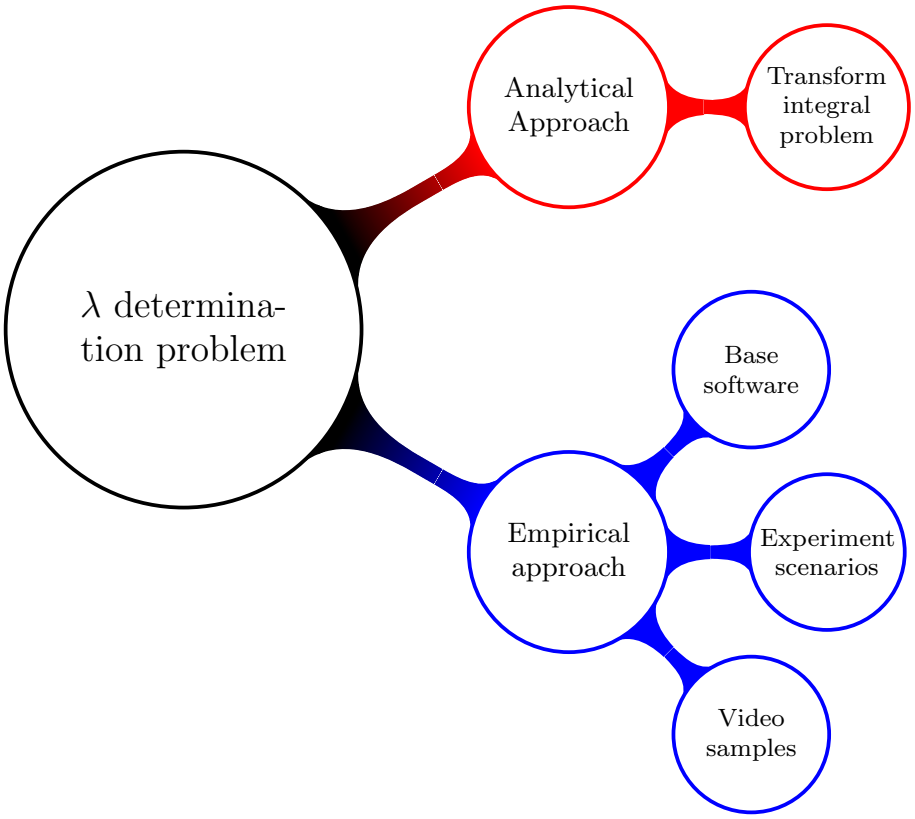
video content and λ optimality[4]. All works used four QP values, possibly because it is the minimum required to calculate the BD-Rate. Finally, it is clear that various works claim the existence of correlations between the movements in the video content and the optimal values for λ. However, such works are far from reaching a consensus on the correlation itself. In fact, the only combination (movement, *m*) that was not claimed to result in λ optimality was ($\downarrow$, $\downarrow$), evidencing the lack of reproducibility for those correlations. Furthermore, the lack of consensus coupled with the lack of video date considered to evaluate those relationships, casts doubts to the very existence of such relationships in the first place.

[4] All of the related works used low resolution video sequences (e.g., CIF).

4 METHOD

Initially, we are going to evaluate the possible approaches for choosing the λ values to be used with the SATD distortion estimation. Figure 6 presents the considered approaches for solving such problem.

Figure 6 – Evaluated approaches for the problem of λ determination.



Source: the author.

There are two paths for solving the problem at hand, the analytical and the empirical. The primary example of analytical approach to such problem comes from the work from Gish and Pierce (1968). Such path seems to be very promising at first, especially because the original formulation already considered a generic average distortion function. A limitation for the method presented in such work is that the cost function was calculated at an element-by-element level (in this context the elements are pixels) before being integrated. This can also be interpreted as the overall distortion being calculated by composing the distortions of individual pixels. However, transforms like the Hadamard cannot be broken down as compositions of smaller transforms as each transformed element depends on all initial elements. Such limitation for the Hadamard is even more evident on the case of transforming a single isolated pixel because the smallest defined Hadamard matrix is of size 2×2. For those reasons, we were not able to further pursue an analytical solution by adapting the method adopted by T. Wiegand and Girod (2001) and diverted to an empirical approach. Still, the empirical

approach required defining the following experimental aspects: (I) the base software to be used (Section 4.1), (II) different scenarios for λ adjustment (Section 4.2) and (III) the set of video sequences to be tested (Section 4.3).

4.1 BASE SOFTWARE

We elected to use the HM (version 16.15) as the basis for the tests featured in this work. The main reason for such decision was that the HM is the reference software for HEVC. Furthermore, all implemented variations were tested with RA configuration from the CTC (BOSSEN, 2012). For the QP, we used the four in $\{22, 27, 32, 37\}$ with QP offset set to "0", i.e., all frames in an encoder execution used the same QP value[1]. The main reason for removing the offset was to have more consistent results for each execution, particularly considering that λ values are a function of QP. Additionally, the CTC configurations use the *HadamardME* flag set to "1" by default. Such flag value means that the SATD is used as distortion metric during FME computation. Meanwhile, the IME employs the SAD regardless of the *HadamardME* flag value.

The BD-Rate (SULLIVAN, G.; BJONTEGAARD, 2001) metric was chosen for quantifying the resulting efficiency differences since it accounts for both distortion (measured in Peak signal-to-noise ratio (PSNR)) and bit rate. The BD-Rate results were gathered by first computing the PSNR for each tested QP. Each PSNR value was derived from the average MSE for the luma channel results from each frame. Such a way of computing the PSNR contrasts to the default HM behavior that computes the PSNR averages directly. The problem with the default implementation in HM is that the PSNR cannot be averaged accurately with a standard arithmetic mean computation because it is a logarithmic metric.

4.2 EXPERIMENTATION SCENARIOS

We performed two sets of experiments evaluating coding efficiency with different scenarios in mind for each set. The first one (Scenario-I) consisted in changing the λ_{motion} for both SAD ($\lambda_{motion,SAD}$) and SATD ($\lambda_{motion,SATD}$). Meanwhile, the second set (Scenario-II) was to evaluate the impact of only changing the $\lambda_{motion,SATD}$.

For Scenario-I, the motion cost equation (Equation 3) was modified, arriving at Equation 27. Those modifications were performed by applying a term (m) multiplying the default λ_{motion} values, as presented in the following equation:

$$J = D_{(SAD|SATD)} + m \times \lambda_{motion,(SAD|SATD)} \times R \tag{27}$$

By making such modification, the cost calculation for both IME and FME are changed with the former using the SAD while the latter employs the SATD. A practical

[1] The removal of QP offsets was also adopted in the empirical set up from F. Zhang and Bull (2018).

way to implement such change is by altering the function denominated *setLambda()*, inside the HM class named *TComRdCost*, as presented in the following code snippet:

```
Void TComRdCost::setLambda(Double dLambda, const BitDepths &bitDepths)
{
  m_dLambda                = dLambda;
  m_sqrtLambda             = sqrt(m_dLambda);
  m_dLambdaMotionSAD[0] = 65536.0 * m_sqrtLambda * lambda_multiplier;
  m_dLambdaMotionSSE[0] = 65536.0 * m_dLambda;

  // FULL_NBIT logic omitted here

  m_dLambdaMotionSAD[1] = 65536.0 * sqrt(dLambda) * lambda_multiplier;
  m_dLambdaMotionSSE[1] = 65536.0 * dLambda;
}
```

In the above code snippet, *lambda_multiplier*[2] corresponds to *m* from Equation 27. Such function was chosen because it sets the λ_{motion} value to be used during ME computation. Furthermore, such function separates the λ setting between the higher precision computations using SSD and the one to be used with either SAD or SATD.

The values set in *setLambda()* are then selected and assigned to *m_motionLambda* in the following function:

```
Void selectMotionLambda(Bool bSad, Int iAdd, Bool bIsTransquantBypass){
        m_motionLambda = (bSad ? m_dLambdaMotionSAD[(bIsTransquantBypass
                && m_costMode==COST_MIXED_LOSSLESS_LOSSY_CODING) ? 1:0]
                + iAdd : m_dLambdaMotionSSE[(bIsTransquantBypass
                && m_costMode==COST_MIXED_LOSSLESS_LOSSY_CODING) ? 1:0]
                + iAdd);
}
```

At first, such one-line function with nested ternary operators may seem confusing. However, the *selectMotionLambda()* function becomes clearer once it is broken down into smaller parts. The first part consists of using *bSad* to select either *m_dLambdaMotionSAD[]* if *bSad* was set to *true* or *m_dLambdaMotionSSE[]* otherwise. In spite of that, by analyzing the calls for *selectMotionLambda()*, *bSad* is always set to *true*, therefore always choosing the *m_dLambdaMotionSAD[]*. Then, remains the expression for which position of *m_dLambdaMotionSAD[]* is selected. In this case, both possibilities were previously covered in *setLambda()*, therefore guaranteeing the use of the modified λ values for *m_motionLambda*. Finally, *m_motionLambda* may be accessed through the *getCost()* and *getCostOfVectorWithPredictor()* functions. Such is the case with the *xTZSearchHelp()* for IME with SAD and *xPatternRefinement()* for FME with SATD. Furthermore, this implementation makes sure that all cost evaluations in ME use the same λ setting logic and multiplication by *m*. Table 4 shows the λ values (from Table 2) that were affected by the described changes for Scenario-I testing as well as the functions that use such values.

[2] The *lambda_multiplier* was added to the base HM allowing us to pass *m* as an execution parameter.

Table 4 – Affected λ values and HM functions for Scenario-I.

λ	Function
$\lambda_{motion,SATD}$	xPatternRefinement
$\lambda_{motion,SAD}$	xTZSearchHelp
	xPatternSearch
	xTZSearch
	xTZSearchSelective
	xMergeEstimation
	predInterSearch
	xCheckBestMVP
	xMotionEstimation

Similarly to Scenario-I, the cost equation in 3 was altered for Scenario-II which lead to Equation 28.

$$J = D_{SATD} + m \times \lambda_{motion,SATD} \times R \tag{28}$$

Despite the similarities between equations 27 and 28 the latter is contained in the former and represents a smaller scope of change to the original HM. That is, Scenario-II consists in varying the Lagrange Multiplier only for the SATD (affecting exclusively the FME) while maintaining the default HM behavior with the SAD. With this test we can verify the adequacy of the standard Lagrange Multiplier for FME as well as draw comparisons to the more generic approach from Scenario-I.

Since this was a smaller change, we simply created a copy of *getCostOfVectorWithPredictor()*, called *getCostOfVectorWithPredictor_Modified()*. Therefore, changing directly the $\lambda \times R$ estimation function with *m_motionLambda* corresponding to *m* as presented in the following code snippet:

```
Distortion getCostOfVectorWithPredictor_Modified(const Int x,const Int y)
{
        return Distortion((m_motionLambda * lambda_multiplier
                * getBitsOfVectorWithPredictor(x, y)) / 65536.0);
}
```

Then, the new function replaced the old one only inside the FME computation function, *xPatternRefinement()*. Since, the FME is set to be performed with the SATD distortion estimation, the described implementation achieves the goal of strictly changing the Lagrange Multiplier values for FME coupled with the SATD.

Both scenarios used the same multiplication values, therefore, having better consistency as well as allowing comparisons between results from different scenarios. We opted to use constant values for *m* to identify the current λ behavior through different ranges without distorting it with another function[3]. Furthermore, by having constant adjustments through different QPs, it is still possible to identify if the results are demanding a different rate weight to QP distribution. Initially we adopted

[3] Because λ is defined in HM as a function of the frame QP, defining *m* as another function would imply having the rate weight as a multiplication of functions $\lambda(QP) \times m(x)$.

m values ranging from 0.2 up to 5.0 by steps of 0.4. Thus, the initial set was: $M = \{0.2, 0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2, 4.6, 5.0\}$. Through analyses of the initial results it was concluded that further refinement was required between 0.2 and 1 by using 0.1 for the step size. Hence, the refinement tests used the following set: $\{0.3, 0.4, 0.5, 0.7, 0.8, 0.9\}$. Furthermore, since we adopted such granularity at values smaller than one, we decided to also add the values in $\{1.1, 1.2, 1.3\}$. It is important to notice that the HM's default behavior was included by adopting $m = 1$. Therefore, the final set M was comprised of 22 values as presented in Equation 29.

$$M = \{0.2, 0.3, \ldots, 1.3, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2, 4.6, 5.0\} \tag{29}$$

Finally, it is also important to notice that, although it is a small change in itself, altering the Lagrange Multiplier in either of the discussed scenarios may have significant implications to the coding process through different mechanisms. Firstly, changing the chosen candidates (references) may affect mode decision choices. Affecting the references can also further impact other blocks through the candidates in inter coding, potentially further changing other references in a snowball effect. Furthermore, in cases with early terminations, changing the $\lambda \times R$ component of the cost may affect how early the computations of candidates are terminated. Therefore, changes to the Lagrange Multiplier may affect (positively or negatively) both the coding time and coding efficiency. We have not addressed the implications to coding time, opting to focus the scope of this work on the coding efficiency implications.

4.3 TEST SEQUENCES

Regarding the video content used for the described tests, we elected to process sequences of lower resolution (when compared to most cases in the CTC). That choice was made so as to maximize the amount of data available and improve the understanding of how the different λ values impact the coding efficiency. Furthermore, we still made sure to vary the resolutions (from *QCIF* up to *4CIF*), fps (from 25 up to 60) and number of frames to be tested (from 112 up to 2101). Therefore, although we lacked in high resolutions tests, we were able to provide a wide range of conditions and a large test data set. Table 5 presents the set of sequences used during the tests[4].

In summary, we considered 47 different sequences, 22 multipliers and 4 QPs, which amounted to 4136 different combinations for each scenario. Hence, all the experiments totaled 8272 HM executions.

We can now add our work to the related works in Table 3, yielding the following table:

[4] All sequences had 4:2:0 chroma sampling.

Table 5 – Tested sequences.

Resolution	Sequence Name	Frame Count	fps
704×576 (4CIF)	city	600	60
	crew	600	
	harbour	600	
	ice	480	
	soccer	600	
352×288 (CIF)	bus	150	30
	city	300	
	crew	300	
	football_b	260	
	harbour	300	
	ice	240	
	soccer	300	
	akiyo	300	29.97
	bowing	300	
	bridge_close	2000	
	bridge_far	2101	
	coastguard	300	
	container	300	
	deadline	1374	
	flower	250	
	foreman	300	
	hall_monitor	300	
	highway	2000	
	husky	250	
	mad900	900	
	mobile	300	
	mother_daughter	300	
	news	300	
	pamphlet	300	
	paris	1065	
	silent	300	
	students	1007	
	tempete	260	
	waterfall	260	
	sign_irene	540	25
352×240 (SIF)	garden	115	29.97
	stefan	300	
	tennis	150	
	tt	112	
176×144 (QCIF)	carphone	382	29.97
	claire	494	
	grandma	870	
	hall_objects	330	
	miss_am	150	
	salesman	449	
	suzie	150	
	trevor	150	

As previously discussed, our work adopted an empirical approach to the problem of Lagrange multiplier optimization considering the scope of the SAD and the SATD on the HM. We took into consideration 22 different constant multiplying factors (more than

Table 6 – Related works summary (including this work).

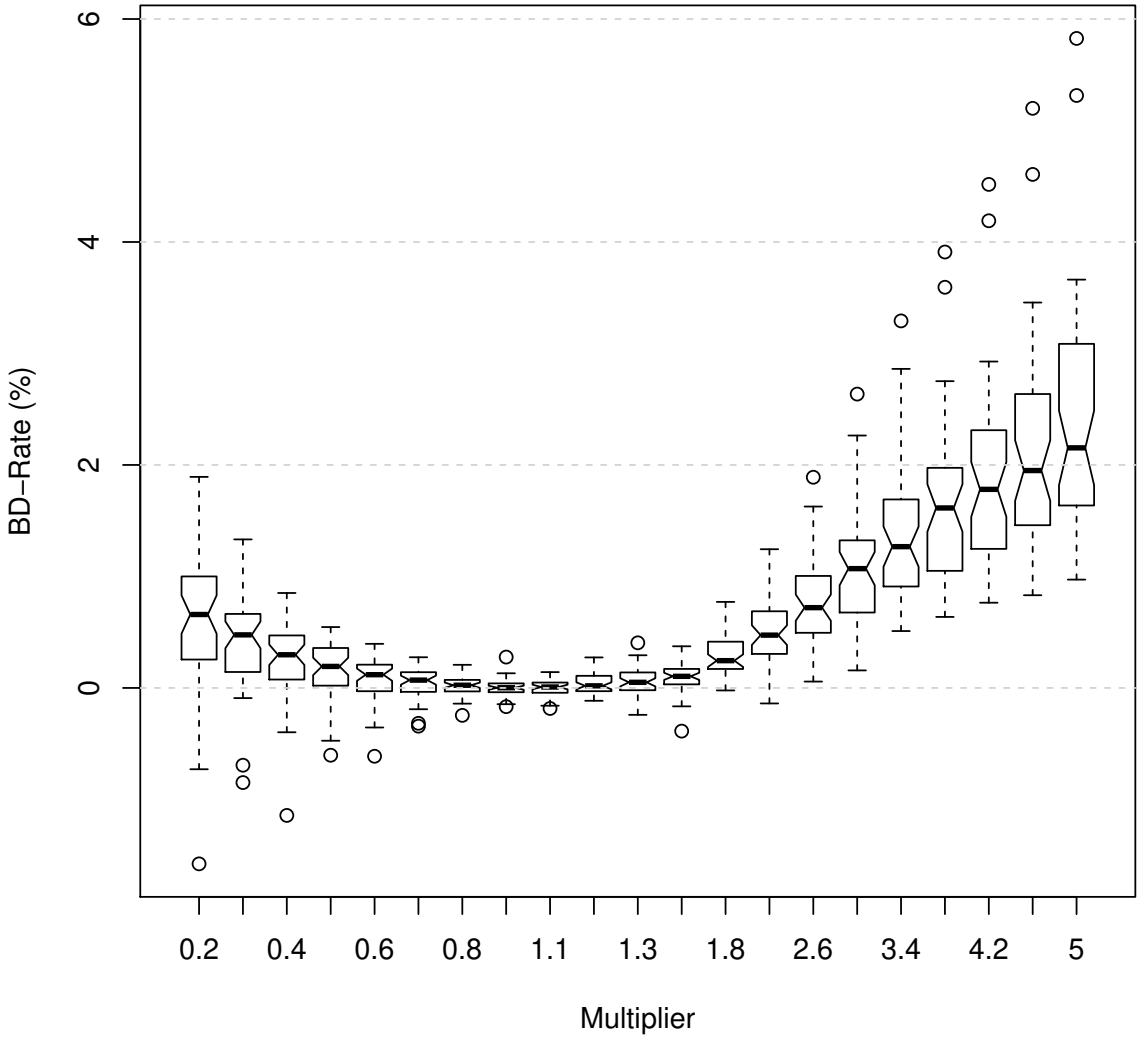
	Scope	Approaches	Reference software		Empirical testing (#)			Adaptive adjustment	
			JM	HM	m	sequences	QPs	movement	m
SANGI; HEIKKILA; SILVEN (2004)	SSD | SAD	Analytical	✓	—	—	—	—	—	—
SYU (2005)	SAD	Analytical	✓	—	—	—	—	—	—
DENG et al. (2013)	SSD | SATD	Analytical	—	✓	—	—	—	—	—
ZHANG, J. et al. (2010)	$\lambda_{motion,SAD}$	Empirical	✓	—	5	3	4	↓ ↑	= ↑
GONZÁLEZ DE SUSO MOLINERO (2016)	$\lambda_{mode,SSD}$ | $\lambda_{motion,SAD}$	Empirical	✓	✓	5	6	4	↓ ↑	↑ =
ZHANG, F.; BULL (2018)	$\lambda_{motion,SAD}$	Empirical	✓	✓	?	9	4	↓ ↑	↑ ↓
This work	$\lambda_{motion,SAD}$ | $\lambda_{motion,SATD}$	Empirical	—	✓	22	47	4	—	—

4 times as much as the numbers adopted in the related works) giving a much more complete picture for the λ behavior. Furthermore, the experiments were done with 47 video sequences (more than 5 times as much as the numbers adopted in the related works) to try and have significantly more video data aiming for better representativeness of video content in overall. Such set of video sequences also provides a better chance of evaluating the relationships between video content and λ optimality. Similarly to the other works, ours took into account four QP values, allowing for the computation of the BD-Rate to evaluate coding efficiency. Finally, as it will be further discussed in Chapter 5, there was no correlation found between video sequence movement and λ optimality, once again evidencing that such relationship might not exist.

5 ANALYSES

In this chapter, we evaluate the results obtained from encoding the 47 video sequences in Table 5 under the two scenarios and four QPs defined in Chapter 4. Figures 7 and 8 present the BD-Rate results for Scenario-I and Scenario-II, respectively.

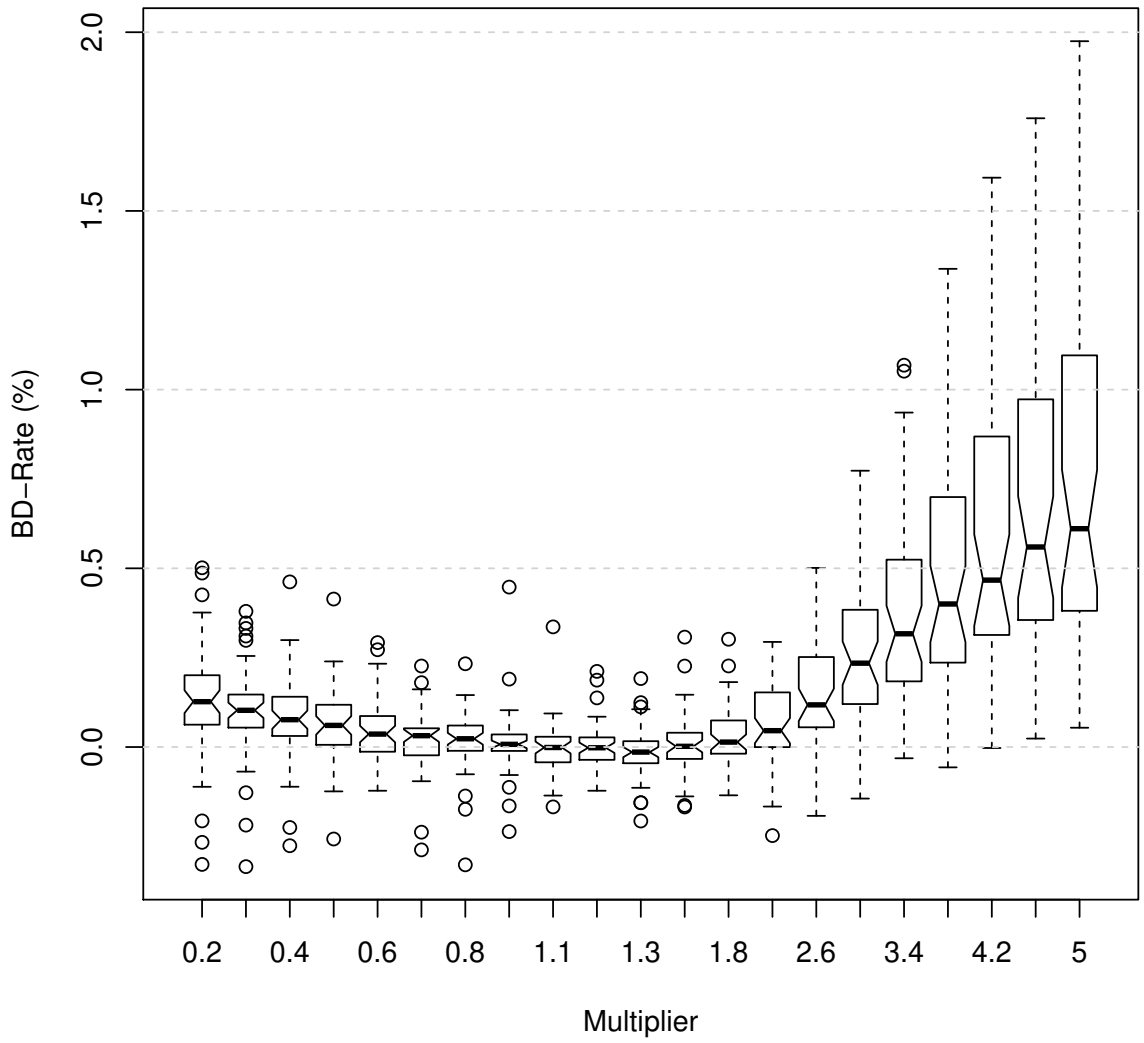
Figure 7 – Boxplot chart with the BD-Rate results when adopting the several multiplication parameters in the ME context (Scenario-I).



Source: the author.

In box plots, the boxes represent the first ($Q_1 = 25\%$) and third ($Q_3 = 75\%$) quartiles. Meanwhile, the second quartile ($Q_2 = 50\%$) is the median which is indicated by a crossing line inside the boxes. The slanted portions of such boxes represent the confidence intervals. Then, out of the boxes, there are whiskers representing the two data points that are furthest away from the median (at each side) but still within

Figure 8 – Boxplot chart with the BD-Rate results when adopting the several multiplication parameters in the FME context (Scenario-II).



Source: the author.

$1.5 \times (Q_3 - Q_1)$. The remaining data points are technically considered discrepancies and are shown as isolated circles.

Each data point in Figures 7 and 8 represents the BD-Rate calculated for a sequence using the four tested QPs (as defined in Section 4.1). Additionally, the results for $m = 1.0$ are implicitly in the charts since they constitute the baseline to which the other multipliers are compared when computing the BD-Rate. Because of that, the results below zero indicate improvements to the base HM and results above zero indicate that the default λs (by using $m = 1$) are a better choice for a given sequence. For instance, with $m \geq 2.6$ in Scenario-I and $m \geq 4.4$ in Scenario-II resulted in worse coding efficiency for all tested sequences. There is also evidence of a sweet spot for

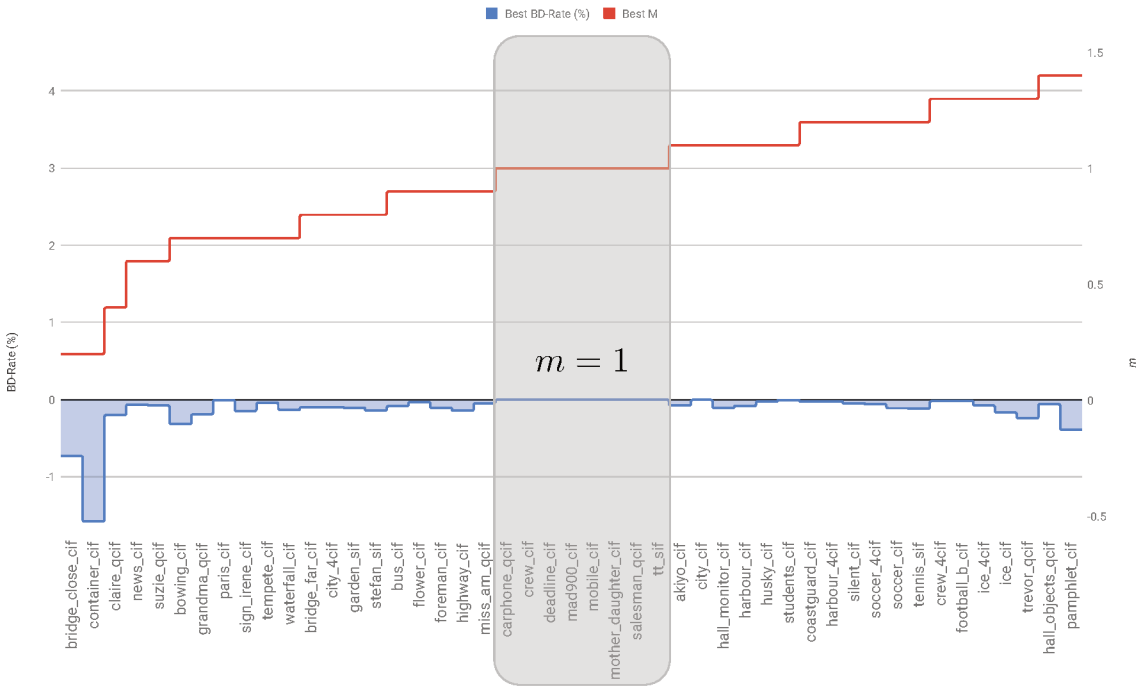
m with values from 0.9 to 1.2 producing similar results in both scenarios. The only multiplier value that produced a noticeably better result within that region of operation was $m = 1.3$ in Scenario-II. Another point that requires evaluation is in regards to the homogeneity of the results. Firstly, the chart for Scenario-I exhibits a wider range for the y-axis because such scenario had a bigger impact on efficiency. Such finding was to be expected since a bigger part of the default coder has been changed. However, there was also a higher amount of discrepancies in Scenario-II with $m < 1$, which was not expected.

To further investigate the initial findings from Figures 7 and 8 it is necessary to break down the boxes and analyze efficiency from a sequence-by-sequence point of view. However, it is not practical to evaluate every multiplier for every sequence especially considering that we are only concerned with the cases yielding the best efficiency results. Because of that we used the BD-Rate figures to select which multiplier produced the best result for each sequence, shown in Figures 9 (Scenario-I) and 10 (Scenario-II). Both figures present in the x-axis the evaluated video sequences sorted by increasing Best M with the shaded area corresponding to the cases that were best coded with the baseline version ($m = 1$). Therefore, all sequences shown to the left of the shaded areas were best coded with $m < 1$ and, similarly, HM provided the best efficiency results with $m > 1$ for the sequences to the right of the shaded portion. The y-axis was split into two for Figures 9 and 10, with the left y-axis presenting the BD-Rate, while the right y-axis shows the Best M values.

The Best M data reinforce the initial evaluation from Figures 7 and 8. In Scenario-I, the best values for m were roughly split between $m < 1$ and $m > 1$ with a few cases being best coded with the baseline multiplier ($m = 1$). Therefore, there is further indication that the baseline multiplier provides the better overall results for Scenario-I. In Scenario-II, there is a clear shift in the Best M regions with $m > 1$ resulting in higher incidence of optimally coded video sequences. Furthermore, Scenario-II can provide optimal coding with higher m values (the highest being $m = 2.2$) in comparison to Scenario-I with an upper bound at $m = 1.4$.

We also sampled the distortion and rates per pixel for each block being processed with the baseline FME. The adoption of average per pixel measures is justified because the blocks being processed vary in size and such averages allowed us to gather data from blocks of different sizes together. The goal for the described sampling was to get a better understanding of each sequence's rate-distortion characteristics at the FME level. It is worth noting that J. Zhang et al. (2010) and F. Zhang and Bull (2018) based their methods on such characteristics. The MSE results were sampled at each 8×8 and 4×4 processed block, including the ones used for composing the distortion of larger or non-square blocks. The rate per pixel estimates were sampled from whole-composed blocks which is the smallest granularity possible for such estimate. Moreover,

Figure 9 – Scenario-I optimal BD-Rate and Best M values for each video sequence, sorted by increasing Best M.



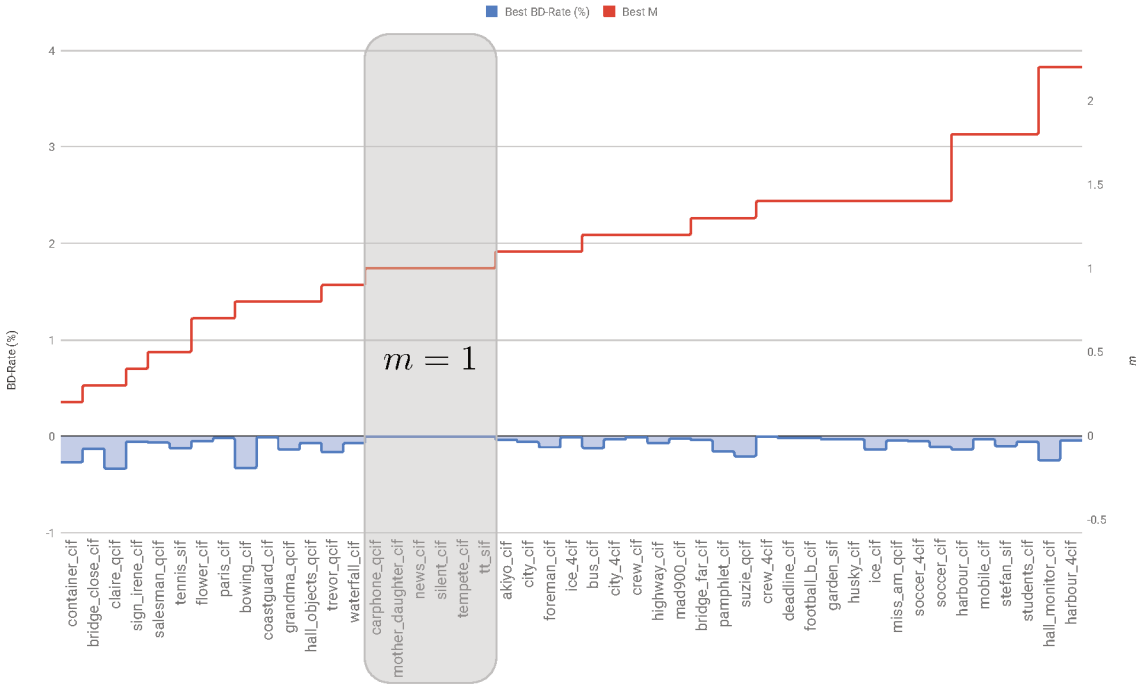
Source: the author.

both the MSE and rate per pixel were relativized to their respective means according to Equation 30, allowing for better visualization of the variations.

$$\text{Relative Deviation} = \frac{sample - mean}{mean} \tag{30}$$

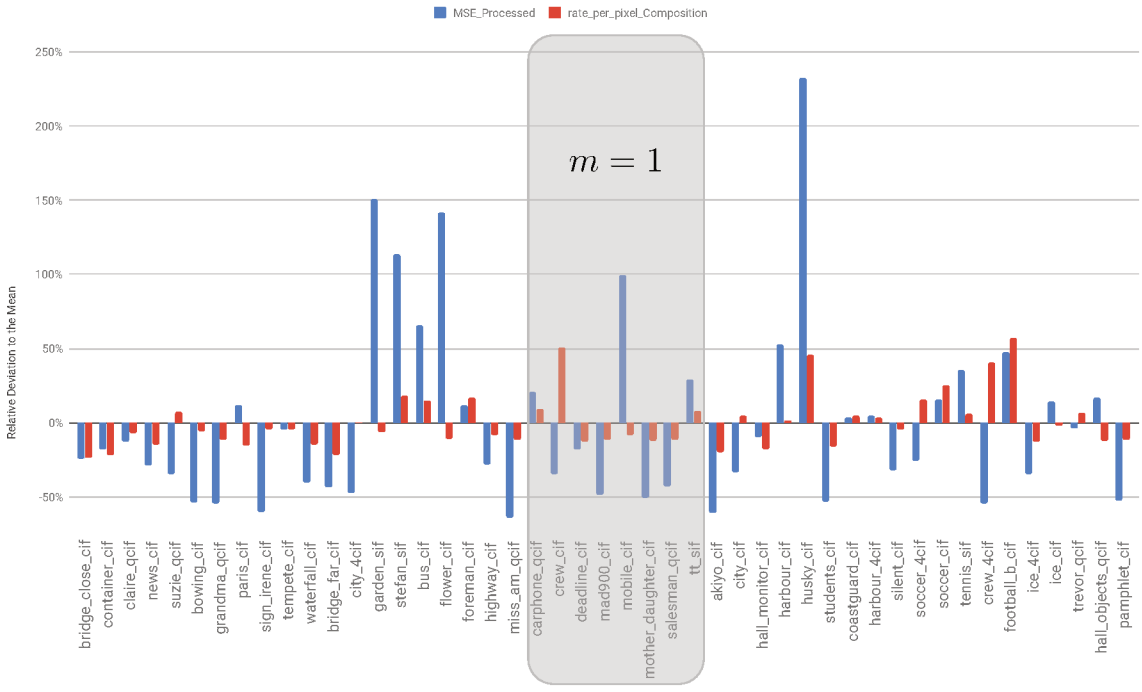
Figures 11 and 12 present the relative deviations for the baseline-sampled MSE and rate per pixel data. Such figures feature the same x-axis sorting and split from Figures 9 and 10. Moreover, the y-axis represents the percent relative deviations to the MSE mean for the MSE (blue bars) and to the rate per pixel mean for the rate per pixel (red bars).

Figure 10 – Scenario-II optimal BD-Rate and Best M values for each video sequence, sorted by increasing Best M.



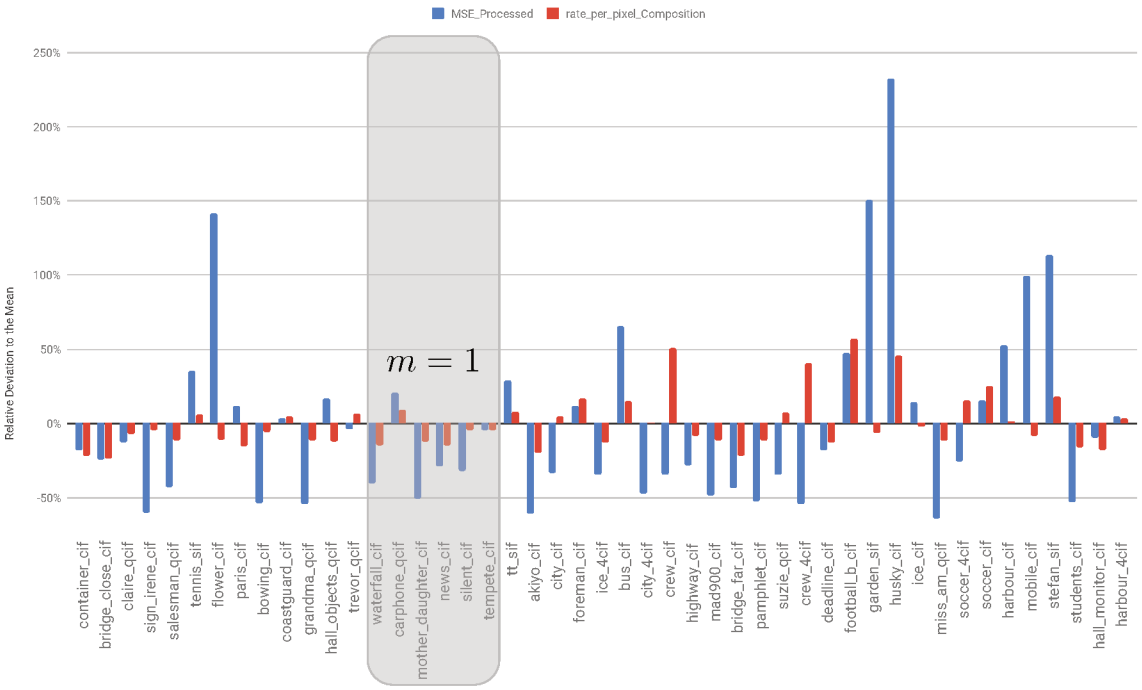
Source: the author.

Figure 11 – Baseline relative MSE and rate per pixel figures sorted by increasing Best M for Scenario-I.



Source: the author.

Figure 12 – Baseline relative MSE and rate per pixel figures sorted by increasing Best M for Scenario-II.



Source: the author.

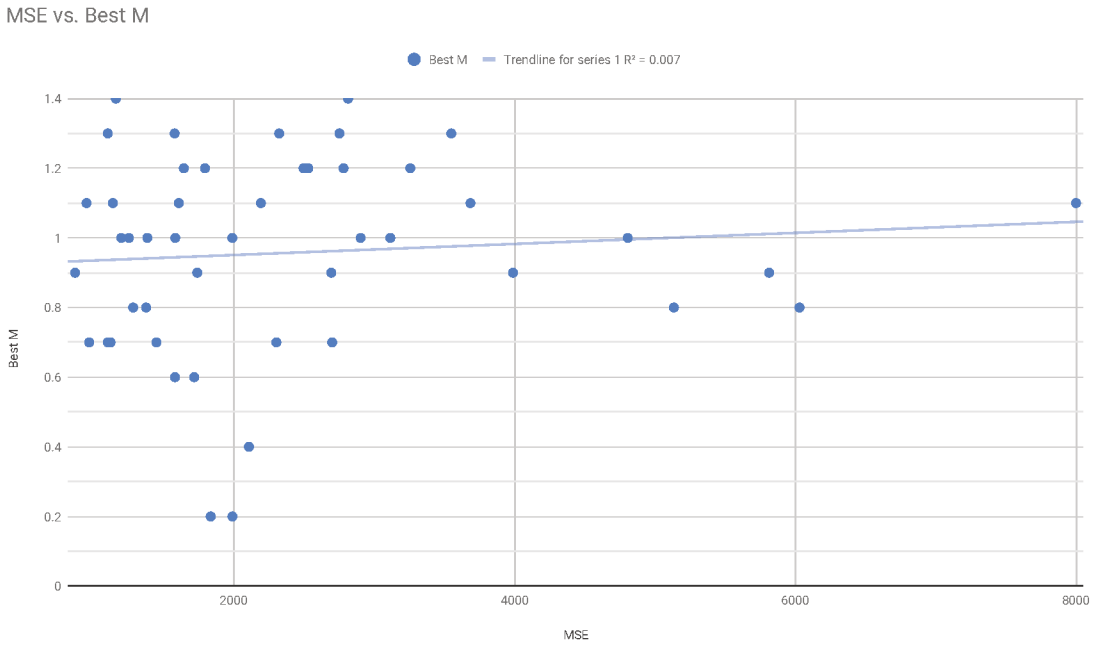
Since the MSE and rate values were sampled during baseline execution, they are the same for the two tested scenarios. The reason for the bars being at different regions in Figures 11 and 12 is because the sequences were ordered by best multiplier. When comparing the relative MSE and rate the former had significantly higher deviations resulting in higher variability with no apparent variables explaining such deviations. Moreover, despite the sequences being ordered by best multiplier, there is no apparent pattern for the distribution of the R-D results.

Still, Figures 11 and 12 do not provide a clear evaluation of potential relationships between the distortion or rate and the best λ multipliers. So, we plotted the sampled MSE against the corresponding best multipliers in Figures 13 and 14. Such figures demonstrate that there was no correlation at all between MSE and the best multipliers on both scenarios and the highest R^2 was 0.043.

We also performed the same evaluation for distortion vs. best multiplier with a different metric, the MATE. Such metric corresponds to the average SATD per pixel (in the same way that the MSE corresponds to the average SSD per pixel). The reason for using a different metric was to make sure the lack of correlation was not exclusive to the MSE. Figures 15 and 16 show no correlation between MATE and best multiplier. In fact, it was even worse with highest R^2 of just 0.033.

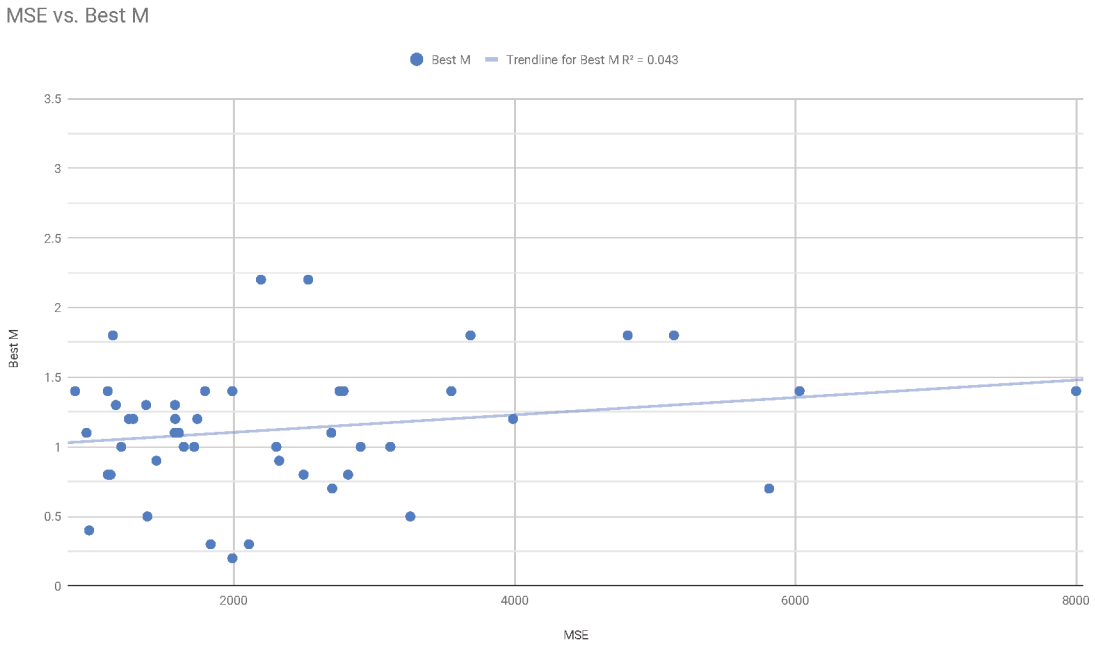
Finally, the figures for distortion vs. best multiplier contradict the conclusions

Figure 13 – MSE vs Best M with linear trendline and the corresponding R^2 for the Scenario-I, identifying if there is a correlation between the two variables.



Source: the author.

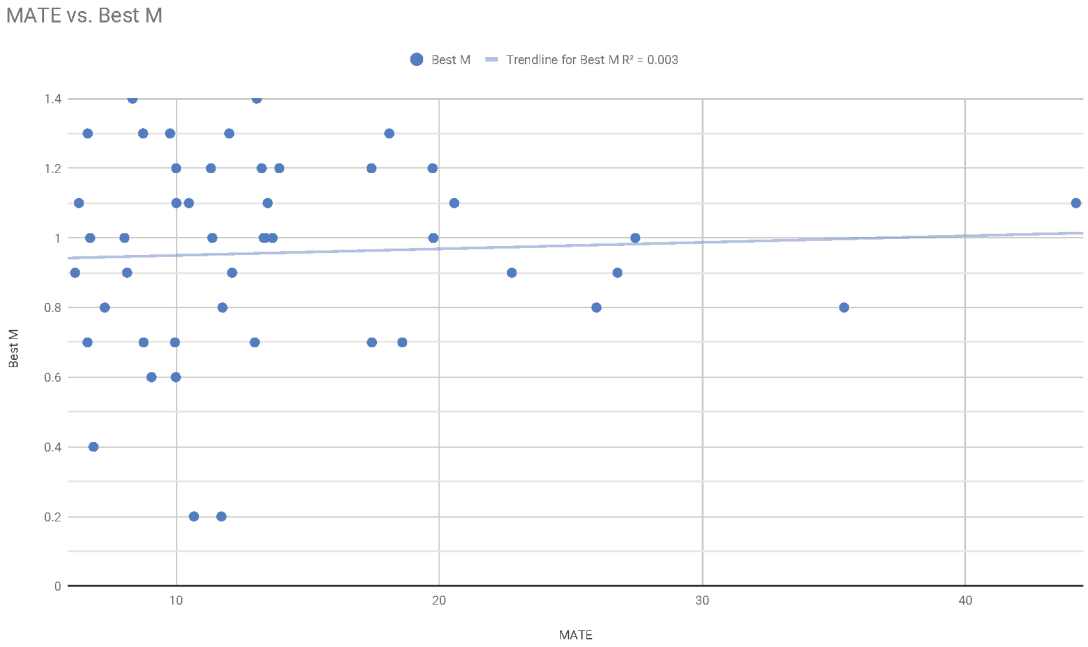
Figure 14 – MSE vs Best M with linear trendline and the corresponding R^2 for the Scenario-II, identifying if there is a correlation between the two variables.



Source: the author.

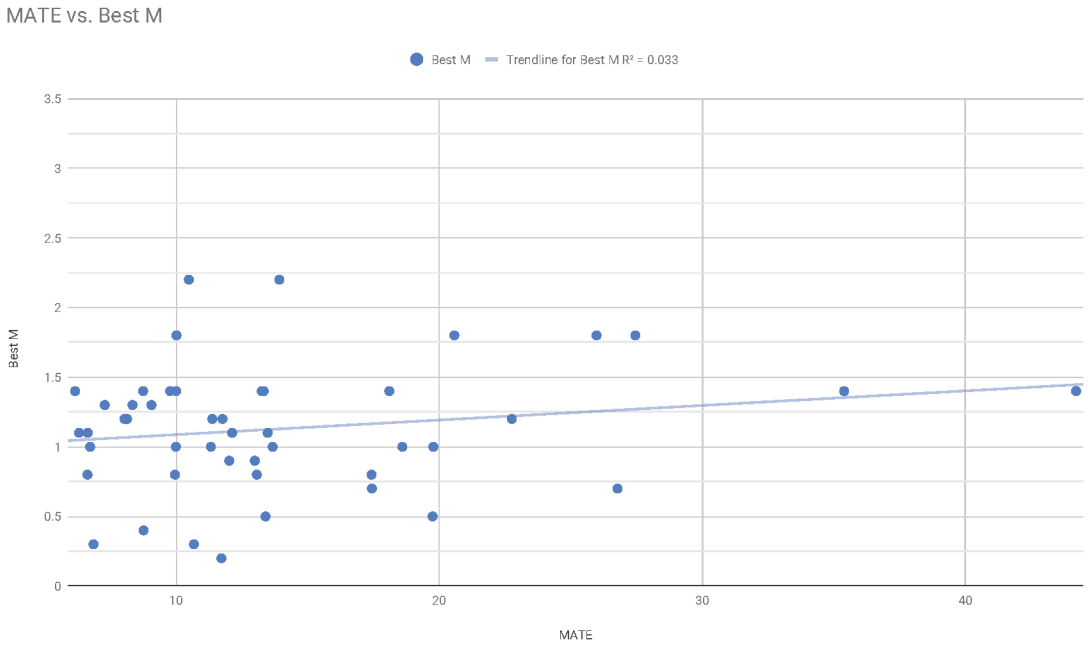
drawn by F. Zhang and Bull (2018) which not only claimed there is a correlation between the two but also used such relationship for their λ adjustment proposal.

Figure 15 – MATE vs Best M with linear trendline and the corresponding R^2 for the Scenario-I, identifying if there is a correlation between the two variables.



Source: the author.

Figure 16 – MATE vs Best M with linear trendline and the corresponding R^2 for the Scenario-II, identifying if there is a correlation between the two variables.

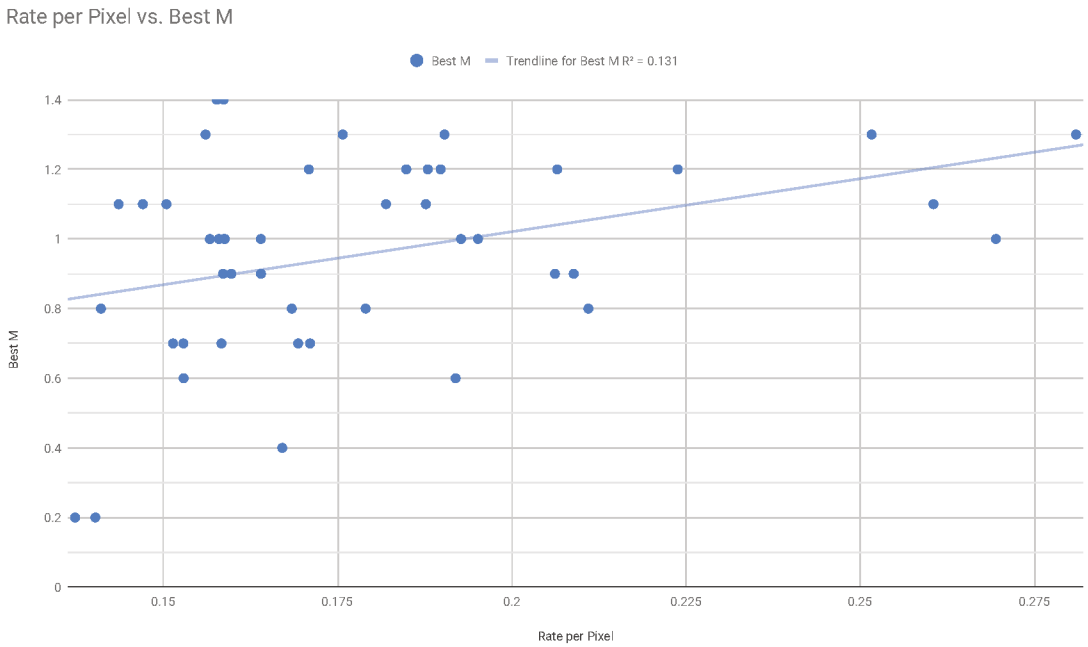


Source: the author.

Similarly to the distortion vs. best multiplier we also analyzed the possibility of a correlation between the rate estimations and the best multipliers. The corresponding

charts are presented in Figures 17 and 18. Once again there was no correlation at all despite the rate being more homogeneous as shown in Figures 11 and 12.

Figure 17 – Rate per pixel vs Best M with linear trendline and the corresponding R^2 for the Scenario-I, identifying if there is a correlation between the two variables.

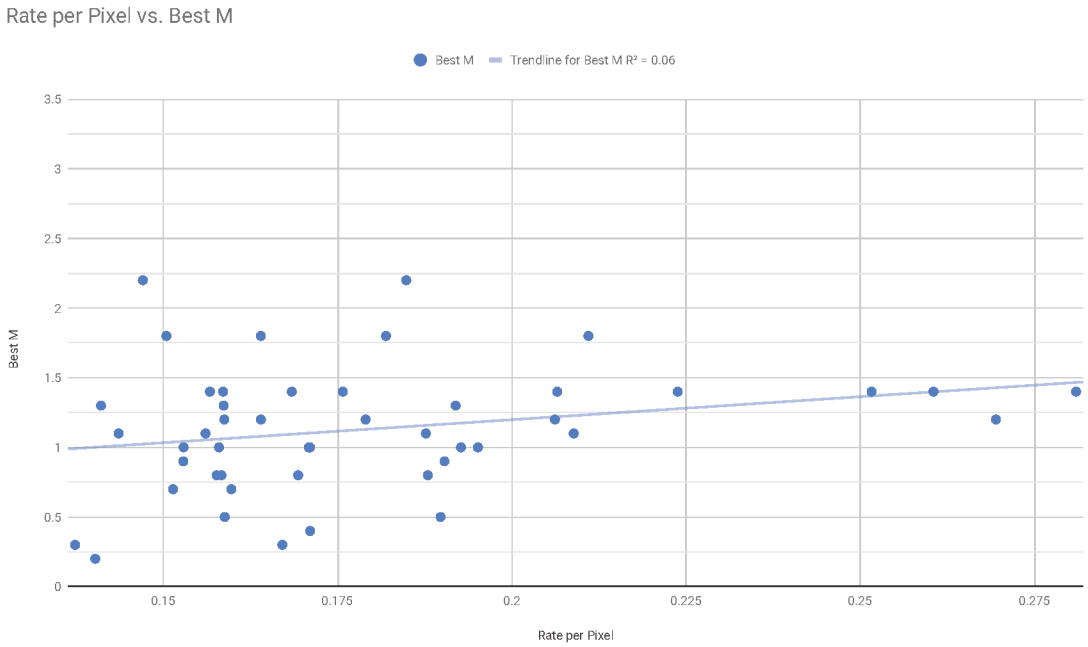


Source: the author.

Finally, we also investigated the claims that the MSE is a good estimator for how dynamic a sequence is. For that we assume the Temporal Information (TI) as being a good measure for video movement. Since the distortions were obtained with the baseline and the TI depends only on the sequence characteristics and the results are the same for both scenarios. Figures 19 and 20 present the TI vs distortion charts. While the former show the distortion as MSE, the latter used the MATE. Although the error was still significant, there is a level of correlation between the sampled distortion and the TI. Furthermore, MSE has better correlation to the TI in comparison to the MATE.

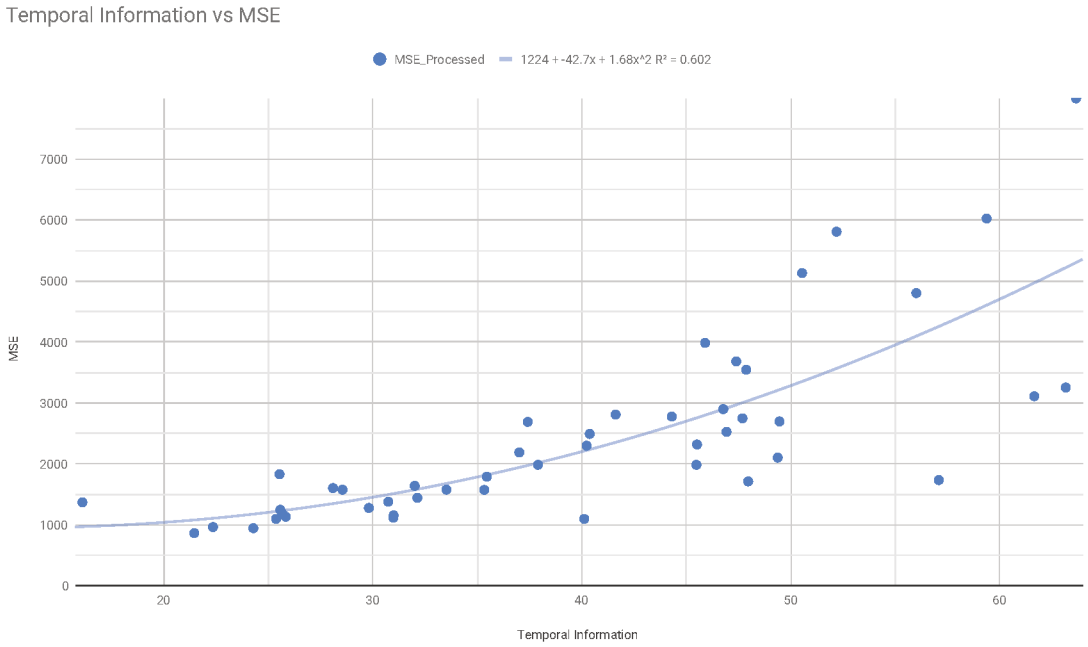
In summary, our findings point to a lack of correlation between the estimated rate and distortion during the FME computation and the overall efficiency results. Additionally, we found evidence of a correlation between the distortion estimations during FME computation and the level of movement in the coded video sequence. Finally, we were able to provide a constant λ multiplicative factor ($m = 1.3$) for the FME cost computation coupled with the SATD which improved coding efficiency considering the tested video sequences, indicating that the FME cost computation is slightly biased towards distortion.

Figure 18 – Rate per pixel vs Best M with linear trendline and the corresponding R^2 for the Scenario-II, identifying if there is a correlation between the two variables.



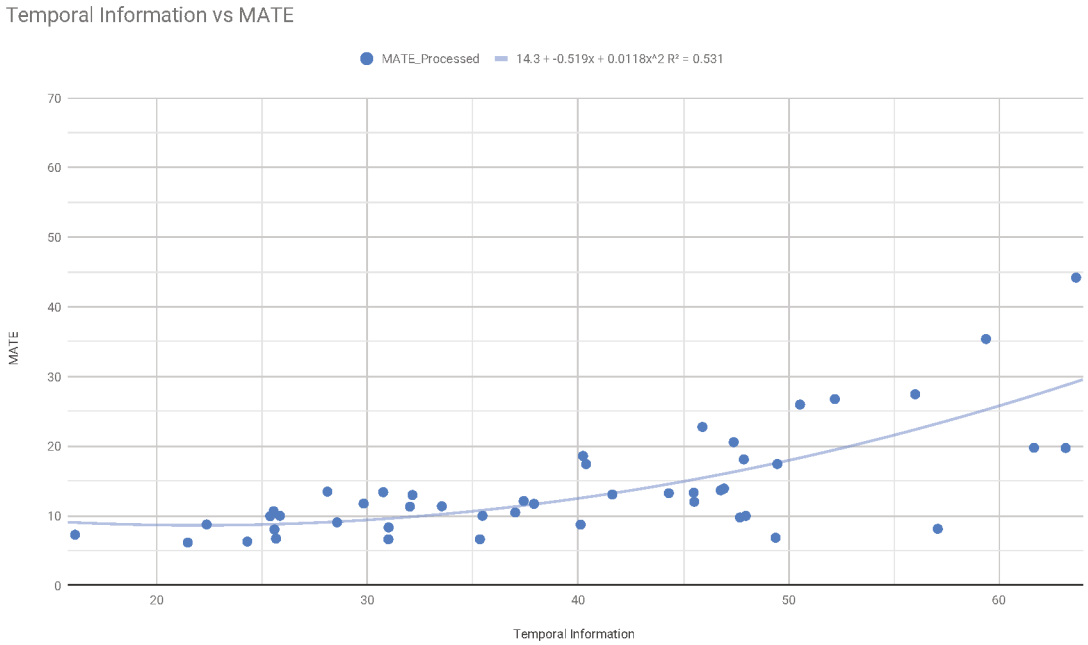
Source: the author.

Figure 19 – TI vs MSE with polinomial degree of 2 trendline and the corresponding R^2, identifying if there is a correlation between the temporal information and the MSE.



Source: the author.

Figure 20 – TI vs MATE with polinomial degree of 2 trendline and the corresponding R^2, identifying if there is a correlation between the temporal information and the MATE.



Source: the author.

6 CONCLUSIONS

Firstly, we were able to relate works from different perspectives starting with the initial proposals for λ multiplier functions. Such works ultimately became the basis of what is implemented in HM (version 16.15). We also identified works that challenged the adopted λ values, with the adaptive adjustment solutions gaining prominence among them. However, there were a few recurrent issues with the related works. The first was the lack of works properly addressing the λ values in the context of the SATD. Furthermore, the works which used empirical data to tune the λ values relied on smaller data sets with few video sequences. Finally, specifically to the work from F. Zhang and Bull (2018), the authors claimed that λ adjustments should be based on how dynamic the video content is.

We considered two approaches for the problem at hand: the analytical and the empirical. The former lead up to the problem of integrating transformed differences, which we were not able to solve.

Then, we followed the investigations through the analytical path, by applying constant multiplicative factors (m) to default λ values. Such investigation was divided in two scenarios:

- Scenario-I where the constant multiplications were performed on an ME level, consequently affecting both RDO computations with SATD in FME and with SAD in other functions using λ_{motion} during ME;
- Scenario-II where the constant multiplications affected solely the FME (which uses the SATD as distortion metric).

With Scenario-I we were able to identify a considerable region where changing m did not produce significantly different coding efficiency. More specifically, multipliers between 0.9 and 1.2 are expected to have similar and satisfactory efficiency results. However, Scenario-II evidenced that using 30% larger λs ($m = 1.3$) during the FME computation **improves the coding efficiency**. That is, the default values for FME are biased towards reducing distortion in detriment of the bit-rate to a point where it is negatively affecting the overall coding efficiency. We found further evidence of such bias when analyzing the best λ for each video sequence where 57.4% of the tested video sequences were best coded with an $m > 1$.

Another important aspect of the empirical testing was to try to find a correlation explaining the best λs for each video sequence. By finding such correlation we would be to adapt the λ to the video sequence and get results closer to the Best M conditions. For that end, we sampled rate and distortion per pixel for blocks being processed in FME. We also computed the TI for each of the tested video sequences. Nevertheless, despite the amount of sampled data and the different metrics considered, we were not

able to find a correlation. Such lack of correlations was not expected especially because F. Zhang and Bull (2018) based their solution on the MSE results. Furthermore, the absence of correlation between TI and Best M questions the claim that the ideal λ values for each video sequence are correlated to how dynamic a sequence is.

One of the limitations from this work regards the resolutions of the tested video sequences. Therefore, further works are needed to investigate if larger resolution videos also lead to similar conclusions. The time necessary for such task may be reduced by using fewer possibilities for *m* based on our findings. Another limitation is that we do not account for interactions between the λs used with the SAD and the ones for SATD. Hence, a possible future work would be to combine the use of different multipliers for each case, greatly increasing the amount of test cases. Finally, there is the possibility that combining more variables together could provide enough information to find a correlation between the variables and the Best M.

REFERENCES

AKRAMULLAH, Shahriar. Video Quality Metrics. In: DIGITAL Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis. Berkeley, CA: Apress, 2014. p. 101–160. ISBN 978-1-4302-6713-3. DOI: 10.1007/978-1-4302-6713-3_4. Available from: https://doi.org/10.1007/978-1-4302-6713-3_4.

BOSSEN, Frank. **Common test conditions and software reference configurations**. Shanghai, Oct. 2012.

DENG, Lei et al. HEVC encoder optimization based on a new RD model and pre-enoding. In: CITESEER. PICTURE Coding Symposium (PCS 2013). IEEE. [S.l.: s.n.], 2013.

GISH, Herbert; PIERCE, John. Asymptotically efficient quantizing. **IEEE Transactions on Information Theory**, IEEE, v. 14, n. 5, p. 676–683, 1968.

GONZÁLEZ DE SUSO MOLINERO, José Luis. **Contributions to the solution of the rate-distorsion optimization problem in video coding**. Leganés: [s.n.], 2016. PhD Thesis.

JCT-VC. **HEVC Test Model**. [S.l.: s.n.], 2013. Available from: http://hevc.hhi.fraunhofer.de/.

JVT. **JM JOINT VIDEO TEAM Reference Software**. [S.l.: s.n.], 2011. Available from: http://iphome.hhi.de/suehring/tml/.

LIM, Keng-Pang; SULLIVAN, Gary; WIEGAND, Thomas. Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods. **Joint Video Team, JVT-R095, Bangkok, Thailand**, 2006.

MCCANN, Ken et al. High efficiency video coding (HEVC) test model 16 (HM 16) improved encoder description. **Joint Collaborative Team on Video Coding, JCTVC-S1002, Strasbourg, FR**, 2014.

RICHARDSON, Iain E. G. **H. 264 and MPEG-4 video compression: video coding for next-generation multimedia**. [S.l.]: John Wiley & Sons Inc, 2003.

RICHARDSON, Iain E. G. **Video codec design: developing image and video compression systems**. [S.l.]: John Wiley and Sons, 2002. ISBN 9780471485537.

SANGI, P.; HEIKKILA, J.; SILVEN, O. Selection of the Lagrange multiplier for block-based motion estimation criteria. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. [S.l.: s.n.], May 2004. p. iii–325. DOI: 10.1109/ICASSP.2004.1326547.

SEIDEL, Ismael. **Redução de Complexidade e Energia em Codificadores de Vídeo Digital Preservando a Eficiência de Codificação: Exploração de Propriedades das Métricas de Distorção Aplicadas no Casamento de Blocos**. 2015. Seminário de Andamento (doutorado) – UFSC, Florianópolis-SC.

SULLIVAN, G. J.; OHM, J. R., et al. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Trans. Circuits Syst. Video Technol.**, v. 22, n. 12, p. 1649–1668, Dec. 2012.

SULLIVAN, G. J.; WIEGAND, T. Rate-distortion optimization for video compression. **IEEE Signal Process. Mag.**, v. 15, n. 6, p. 74–90, Nov. 1998.

SULLIVAN, Gary J. Overview of International Video Coding Standards (preceding H.264/AVC). In:

SULLIVAN, Gary; BJONTEGAARD, Gisle. Recommended simulation common conditions for H. 26L coding efficiency experiments on low-resolution progressive-scan source material. **ITU-T VCEG, Doc. VCEG N**, v. 81, p. 2001, 2001.

SYU, Eric. **Implementing rate-distortion optimization on a resource-limited H. 264 encoder**. 2005. PhD thesis – Massachusetts Institute of Technology.

SZE, Vivienne; BUDAGAVI, Madhukar; SULLIVAN, Gary J. High efficiency video coding (HEVC). **Integrated circuit and systems, algorithms and architectures**, Springer, p. 1–375, 2014.

WIEGAND, T.; GIROD, B. Lagrange multiplier selection in hybrid video coder control. In: PROCEEDINGS 2001 International Conference on Image Processing (Cat. No.01CH37205). Thessaloniki, Greece: [s.n.], 2001. 542–545 vol.3. DOI: 10.1109/ICIP.2001.958171.

ZHANG, F.; BULL, D. R. Rate-distortion Optimization Using Adaptive Lagrange Multipliers. **IEEE Transactions on Circuits and Systems for Video Technology**, p. 1–1, 2018. ISSN 1051-8215. DOI: 10.1109/TCSVT.2018.2873837.

ZHANG, J. et al. Context Adaptive Lagrange Multiplier (CALM) for Rate-Distortion Optimal Motion Estimation in Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 20, n. 6, p. 820–828, June 2010. ISSN 1051-8215. DOI: 10.1109/TCSVT.2010.2045915.

ZHAO, Long et al. A background proportion adaptive Lagrange multiplier selection method for surveillance video on HEVC. In: IEEE. 2013 IEEE International Conference on Multimedia and Expo (ICME). [S.l.: s.n.], 2013. p. 1–6.