UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

João Mai

Coding Efficiency Analysis of Fractional Motion Estimation and Affine Motion Estimation in VVC

Florianópolis
2024

João Mai

Coding Efficiency Analysis of Fractional Motion Estimation and Affine Motion Estimation in VVC

Monografia de Trabalho de Conclusão de Curso do Curso de Graduação em Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.
Orientador: Prof. José Luís Almada Güntzel, Dr.

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Mai, João
Coding Efficiency Analysis of Fractional Motion Estimation and Affine Motion Estimation in VVC / João Mai ; orientador, José Luís Güntzel, 2024.
38 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2024.

Inclui referências.

1. Ciências da Computação. 2. Affine. 3. FME. 4. VVC. 5. Codificação de vídeo. I. Güntzel, José Luís. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

João Mai

Coding Efficiency Analysis of Fractional Motion Estimation and Affine Motion Estimation in VVC

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel em Ciências da Computação" e aprovado em sua forma final pelo Curso de Graduação em Ciências da Computação.

Florianópolis, 18 de novembro de 2024.

Profa. Lúcia Helena Martins Pacheco, Dra.
Coordenador(a) do Programa de Graduação
em Ciências da Computação

Banca Examinadora:

Prof. José Luís Almada Güntzel, Dr.
Universidade Federal de Santa Catarina

Prof. Ismael Seidel, Dr.
Universidade Federal de Santa Catarina

Prof. Wyllian Bezerra da Silva, Dr.
Universidade Federal de Santa Catarina

Florianópolis, 2024

Dedicado esse trabalho aos meus gatos Pandora e Jojo, que me acompanharam durante noites tardias de escrita quando todos estavam dormindo.

AGRADECIMENTOS

Agradecimentos aos meus colegas de labaratório e professores, pela ajuda direta e indireta na realização deste trabalho.

Preciso agradecer também meus amigos, que me acompanham em viagens, bares e sala de aula.

E por fim agradeço também minha família, especialmente meu irmão Pedro, que precisa me aturar desde sempre.

"O tempo passa rápido quando a gente se diverte"
(Um amigo do André)

RESUMO

O padrão Versatile Video Coding (VVC) surgiu para contornar as exigências cada vez maiores impostas pelo consumo de vídeo em alta resolução. O VVC melhora ferramentas bem estabelecidas, como a Fractional Motion Estimation (FME), além de introduzir novas, como a Affine Motion Estimation (AME), aumentando a eficiência de codificação em comparação ao seu antecessor, o High Efficiency Video Coding (HEVC). Trabalhos relacionados demonstraram que a eficiência de codificação da FME foi reduzida no VVC Test Model (VTM) em comparação ao HEVC Test Model (HM). Assim, é hipotetizado que a AME possa estar interagindo com a FME, uma vez que a AME também depende da geração de amostras interpoladas em possíveis posições fracionárias. Portanto, este trabalho analisa, utilizando Bjøntegaard Delta Rate (BD-Rate), a eficiência de codificação de ambas as ferramentas, FME e AME, considerando suas implementações no VTM e duas configurações, Random Access (RA) e Low Delay (LD). Também mostra-se que desabilitar a AME reduz a eficiência média de codificação em 2,48% (RA) e 3,73% (LD), enquanto desabilitar a FME reduz em média 0,63% (RA) e 0,93% (LD), confirmando o menor impacto da FME. Ao desabilitar ambas as ferramentas, a eficiência média de codificação é reduzida em 4,19% (RA) e 5,05% (LD), valores superiores à soma das perdas individuais de eficiência de codificação, confirmando um pequeno, mas presente, efeito compensatório da AME.

Palavras-chave: *Affine*. FME. VVC. Codificação de vídeo.

ABSTRACT

The Versatile Video Coding (VVC) standard emerged to circumvent the ever-higher demands imposed by high-resolution video consumption. VVC improves upon well-established tools, such as the Fractional Motion Estimation (FME), as well as brings new ones, such as the Affine Motion Estimation (AME), increasing the coding efficiency when compared to its predecessor, the High Efficiency Video Coding (HEVC). Related work demonstrated that the overall coding efficiency of the FME was reduced in the VVC Test Model (VTM) compared to HEVC Test Model (HM). Thus, it is hypothetised that the AME may be interacting with FME once AME also relies on the generation of interpolated samples at possible fractional positions. Therefore, this work analyzes, using Bjøntegaard Delta Rate (BD-Rate), the coding efficiency of both tools, FME and AME, considering their implementation in the VTM and two configurations, Random Access (RA) and Low Delay (LD). This work shows that disabling the AME reduces the average coding efficiency by 2.48% (RA) and 3.73% (LD) while disabling the FME reduces, on average, 0.63% (RA) and 0.93% (LD), confirming the lower impact of FME. When disabling both tools, the average coding efficiency is reduced by 4.19% (RA) and 5.05% (LD), which are higher than the sum of the individual coding efficiency losses, confirming a small but present AME compensatory effect.

Keywords: Affine. VVC. FME. Video coding.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AME	Affine Motion Estimation
AMVR	Adaptive Motion Vector Resolution
BD-Rate	Bjøntegaard Delta Rate
BMA	Block Matching Algorithm
CLI	Command Line Interface
CP	Control Point
CTC	Common Test Conditions
FHD	Full High Definition
FME	Fractional Motion Estimation
HD	High Definition
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
IME	Integer Motion Estimation
JVET	Joint Video Experts Team
LD	Low Delay
LDP	Low Dealy with P-slices only
ME	Motion Estimation
MSE	Mean Squared Errors
MV	Motion Vector
MVP	Motion Vector Prediction
PMV	Predicted Motion Vector
PSNR	Peak Signal-to-Noise Ratio
QP	Quantization Parameter
RA	Random Access
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
SD	Standard Definition
SIMD	Single Instruction Multiple Data
TME	Translational Motion Estimation
UHD	Ultra High-Definition
VTM	VVC Test Model
VVC	Versatile Video Coding
WQVGA	Wide Quarter Video Graphics Array

CONTENTS

1 INTRODUCTION

The last decade saw substantial improvements in Internet connections worldwide, which, paired with the popularity of smartphones and other portable devices, resulted in a significant increase in digital video consumption. In 2023, 54% of data volume on non-portable devices (such as computers and smart TVs) corresponded to on-demand streaming (platforms like YouTube and Netflix), with 57% for the same category on mobile devices (SANDVINE, 2024).

In the face of a demand for 4K (3840×2160 pixels per frame) and higher resolutions, more efficient compression standards are made necessary, employing new tools and methods capable of enabling the transmission and storage of videos at those higher resolutions. In this context, the Joint Video Experts Team (JVET) introduced the Versatile Video Coding (VVC) standard (ISO CENTRAL SECRETARY, 2020) in 2020, succeeding the High Efficiency Video Coding (HEVC) (ISO CENTRAL SECRETARY, 2013).

When comparing the reference software of the two standards, VVC Test Model (VTM) and HEVC Test Model (HM), the former brings an average 38.9% Bjøntegaard Delta Rate (BD-Rate) improvement in coding efficiency for Random Access (RA) configuration and 30.9% for Low Delay (LD) configuration, however, with coding time increases of 803% and 659%, respectively (BROSS *et al.*, 2021). The increase in coding efficiency and coding time in VTM is attributed to the higher complexity of VVC, which not only enhances well-established tools, such as the Fractional Motion Estimation (FME), but also adopts new ones, such as the Affine Motion Estimation (AME) (BROSS *et al.*, 2021; YANG *et al.*, 2021).

While the FME tool in the HM was one of the culprits for inter-prediction high complexity, the FME in VTM corresponds to a smaller percentage of the overall runtime, with the average measurement being 60% of its predecessor (SIQUEIRA; CORREA; GRELLERT, 2021), which is partially explained by Single Instruction Multiple Data (SIMD) optimizations. Moreover, fully disabling the FME in HM degrades the coding efficiency by 10.89% and 12.75%, average BD-Rate, for RA and LD configurations, respectively (SEIDEL, 2019). Yet, the cost of disabling the FME in VTM is smaller, with average BD-Rate of 0.67% (RA) and 1.21% (Low Dealy with P-slices only (LDP)) (FILHO *et al.*, 2021).

The adoption of AME increases the coding time of VTM by 27%, on average (YANG *et al.*, 2021), reducing the rate for Motion Vector (MV) signaling and thus improving coding efficiency by 3.4%. In fact, given that AME produces a block-based motion field with 1/16-precision MVs, the hypothesis is that the AME duplicates certain aspects of the FME, thus partially justifying the lower coding efficiency impact of FME in VTM.

This work aims to provide an overview in regards to the inter-prediction step in VVC, analyzing the FME and AME tools impact on coding efficiency. The goal is to better understand the trade-off between compression and complexity associated with each tool, shedding light to the conundrum that is introducing higher complexity tools without understanding how they interfere with each other, as most works tend to focus on tools in isolation.

1.1 GOALS

The main goal of this work is to conduct a comparative analysis of the contribution of the FME and AME tools to the coding efficiency of the VVC. This analysis will be carried out through the evaluation of BD-Rate data obtained by coding different video sequences in various configurations using a modified version of VVC's reference software, the VTM.

1.1.1 Specific Goals

- Quantify the impact of FME and AME on the coding efficiency of different sequences present in the Common Test Conditions (CTC) for the RA and LD temporal configurations, using BD-Rate as the metric;

- Evaluate if and to what extent the AME can compensate for the lack of FME during coding.

1.2 STRUCTURE

This work is organized as follows:

- **Background**, introducing concepts related to video coding in general, specific concepts related to the VVC standard, and the metrics used for analysis;

- **Related work**, briefly reporting the results of works with overlapping goals to this one;

- **Method**, describing what was done in order to achieve the results;

- **Results**, presenting and analyzing the data obtained;

- **Conclusion**, discussing the results and mentioning possible future works.

2 BACKGROUND

This chapter presents an overview of video coding concepts, including the structure of a video encoder and its inner workings, focusing on the inter-frame prediction step, especially the FME and AME, as well as objective metrics used for comparison between encoded sequences.

2.1 BASIC CONCEPTS

A video can be defined as a sequence of frames (still images) presented in a specific order, where each frame is represented as a matrix of picture elements, commonly known as pixels. One of the simplest forms of image is the grayscale form, such as Figure 1, where each pixel represents a brightness or luminance (luma) value.

Figure 1 – Example of a grayscale image, 8 *bits* per pixel.

Source: author.

In order to represent colors in an image, a color system is necessary. Color systems are models that represent colors of an image using multiple channels, *e.g.*, the RGB model, where there are three channels representing the intensity of the primary colors red, blue, and green. The RGB system usually has 8 bits per channel, totaling 24 bits per pixel. Information from just one channel of a pixel is called a sample.

Two other major characteristics of a video are the spatial resolution and temporal resolution. Spatial resolution refers to the frame size, *i.e.*, Standard Definition (SD) (720 × 480) or High Definition (HD) (1280 × 720), while temporal resolution refers to the frame rate, *i.e.*, 30 frames per second, usually presented as the amount of frames in 1 second of video.

Using the information of how many bits per pixel there are, resolution and frame rate, it is possible to compute the bitrate for a raw (uncompressed) sequence:

$$bitrate_{raw} = \text{bits per pixel} \times \underbrace{\text{width} \times \text{height}}_{\text{frame size}} \times \text{frame rate} \tag{1}$$

If the duration is also known for the sequence, it is also possible to calculate the raw file size:

$$size_{raw} = bitrate_{raw} \times \text{duration} \tag{2}$$

A sequence using 24 bits per pixel for the color representation, Full High Definition (FHD) resolution and 24 frames per second, has a bitrate of 1.2 Gb/s, around 150 MB/s. Considering a 1 hour sequence, the file size in disk would be 537 GB. It is noted that those are somewhat prohibitive values when considering the real world, showing that video compression is necessary for the practical use of video in the everyday life, being it storing a video file locally or watching it through a mobile application.

In order to compress a sequence, exploiting redundancies found within it is necessary. According to Shi and Sun (2019), it is possible to classify redundancies in the following:

- **Psychovisual**: redundancies found exploiting the human visual system;
- **Statistical**: redundancies associated with the symbols of a sequence, which are split in following categories:
  - **Interpixel**: pixels within a frame or sequence are correlated, being categorized as:
    - **Spatial**: represents the correlation found in pixels within a given frame, also known as intra-frame redundancy;
    - **Temporal**: represents the correlation found in pixels from neighboring frames, also known as inter-frame.
  - **Coding**: correlation found in the information medium of representation.

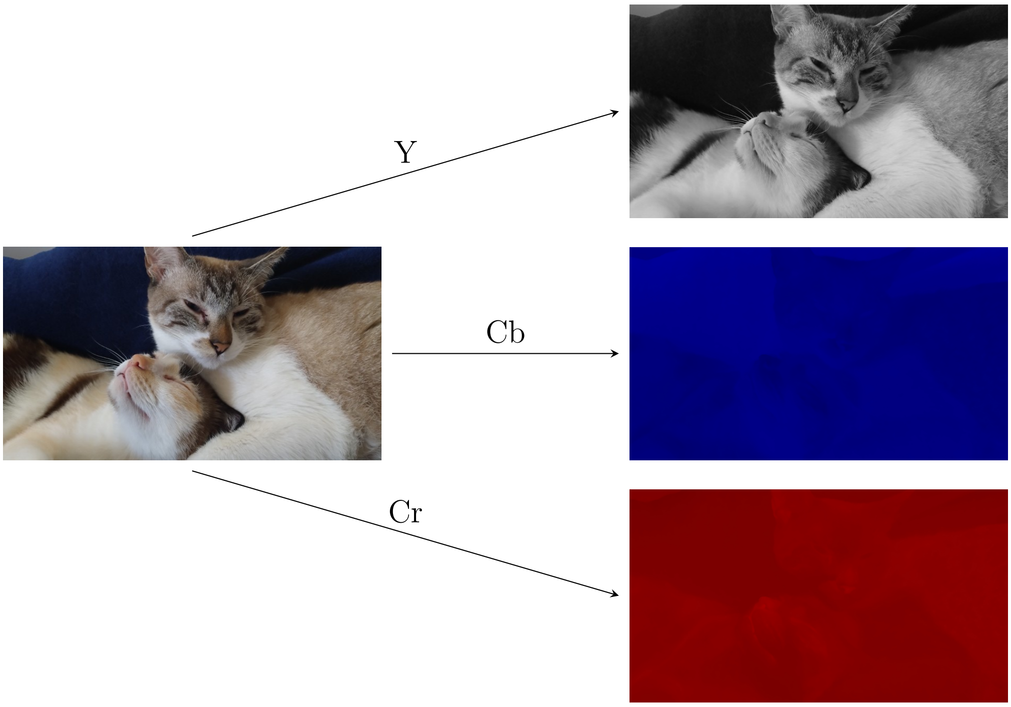
2.1.1 Psychovisual redundancy

Psychovisual redundancies are characterized by information that the human visual system is not sensitive enough to capture. Due to the physical characteristics of the visual system, small variations in luminance (brightness) of an image are more easily perceived than variations in color (HUNT, 2005).

To exploit this characteristic, a color space that separates light and color information in the image is needed, such as the YCbCr space. In this color space, Y corresponds to the luminance component (luma), while Cb and Cr correspond to the chrominance components (chroma) for blue difference and red difference, respectively. There is also a green difference chroma component (Cg), which can be derived from the others and is therefore often omitted. Figure 2 shows an example of the frame representation for the luma, blue- chroma and red-difference components.

Using the YCbCr color space, it is possible to apply a chroma subsampling technique, enabling a reduction to chroma information without significant loss of subjective

Figure 2 – Example of frame representation for the three YCbCr channels.

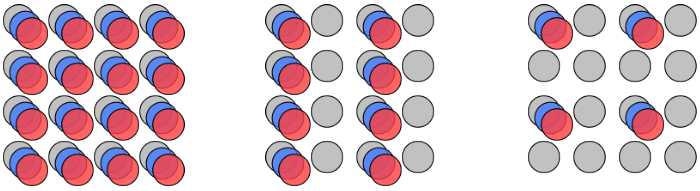


Source: author.

quality of the image. To perform subsampling, vertical or horizontal lines from the chroma matrix of an image are eliminated, with the retained samples being interpolated during decoding.

Figure 3 illustrates possible representations of the YCbCr color space, with 4:4:4 having no subsampling; 4:2:2 with chroma samples removed every 2 columns; and 4:2:0 having chroma samples removed every 2 lines and columns.

Figure 3 – Examples of chroma subsampling.



(a) 4:4:4 (b) 4:2:2 (c) 4:2:0

Source: Adapted from Seidel (2014).

Considering that each channel uses 8 bits per pixel, for a group of 4 pixels, 128 bits are used in the 4:4:4 subsampling, while 56 bits are used in 4:2:0, resulting in a 50% reduction. The use of subsampling introduces minimal visual losses while providing a significant gain in lossy compression.

2.1.2 Statistical redundancy

Statistical redundancies are related to the repetition of symbols within a sequence. According to Shi and Sun (2019), they can be classified as inter-pixel redundancy and coding redundancy. Coding redundancy is not exclusive to video compression; it can be exploited in the compression of any type of data. For this, entropy coding is performed, compressing data by modifying its representation: instead of using a fixed length for the number of bits in the representation, a variable length is used based on the probability of occurrence of each symbol.

Considering Table 1, the average number of bits per symbol is originally 2, while using variable-length coding, the average is 1.4 bits per symbol (calculated in Equation 3), which is 70% of the original size.

Table 1 – Example of symbol distribution.

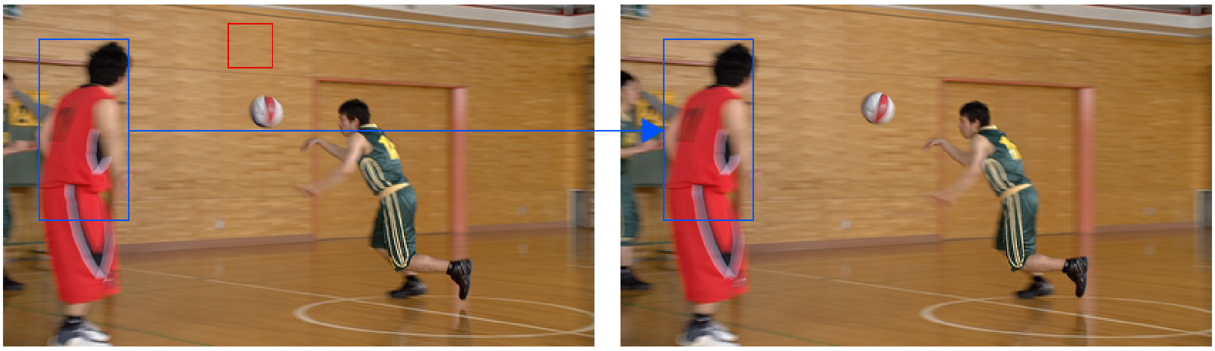
Symbol	Probability	Original code	New code
a	70%	00	0
b	20%	01	10
c	7%	10	110
d	3%	11	111

Source: author.

$$L = 0.7 \times 1 + 0.2 \times 2 + 0.07 \times 3 + 0.03 \times 3 = 1.4 \tag{3}$$

Another kind of statistical redundancy is inter-pixel redundancy, where the redundancy among pixels in the same frame (intra-frame) or across different frames (inter-frame) is found. Figure 4 illustrates both cases: in the first frame, there is spatial redundancy in the area of the red square, where the pixels within that area are very similar; the blue rectangles indicate temporal redundancy, where neighboring frames show a very similar image of the basketball player.

Figure 4 – Example of inter and intra redundancies.



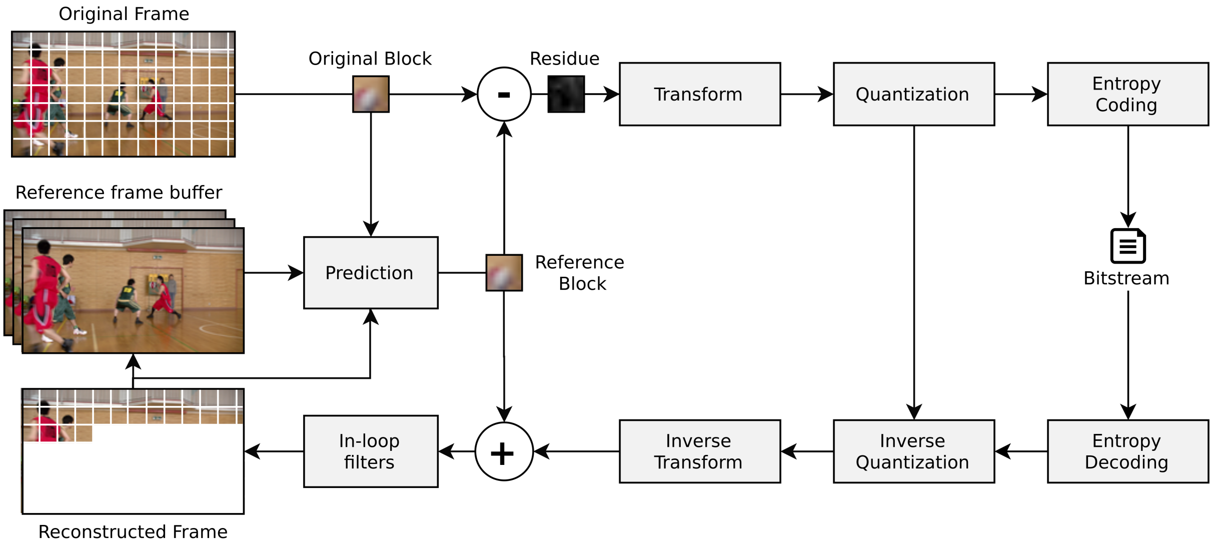
Source: author.

To exploit inter-pixel redundancies, prediction must be performed during the coding loop. Given the frame to be predicted, called the original frame, a reference frame that is similar to the original frame is found and the residuals between the two frames (the pixel-by-pixel difference of one of the channels) are computed. It is expected that the residuals will have a lower entropy compared to the original frame, allowing for more efficient coding. On the decoder side, the original frame is reconstructed using the reference frame and the residuals.

2.2 HYBRID CODING MODEL

Figure 5 presents a simplified model of a hybrid video encoder. The encoder combines the steps of prediction, transform, and quantization into a single model, hence the name hybrid video encoder. Since its introduction with the H.261 standard in 1988, this has been the model used by so-called hybrid encoders ever since (RICHARDSON, 2003).

Figure 5 – Simplified model of a hybrid video encoder.

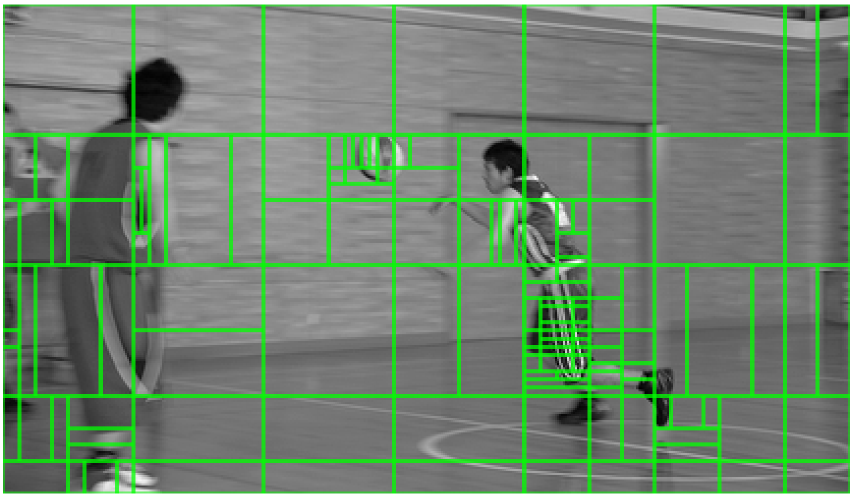


Source: Filho (2022), adapted from Richardson (2003).

To allow for greater versatility and increased compression, standards also adopt partitioning of frames into rectangular blocks, enabling each step of the encoder to be performed on the blocks rather than the entire frame. Figure 6 illustrates an example of this partitioning. In the VVC standard, this partitioning is performed for each channel, with blocks ranging from 128 × 128 to 4 × 4 for luma samples, and 64 × 64 to 2 × 2 for chroma samples (HUANG *et al.*, 2021).

The prediction step is performed for each original block, where a decision must be made whether to use intra prediction, which looks for a reference block within the same

Figure 6 – Example of Y channel partitioning for a frame in the VTM.

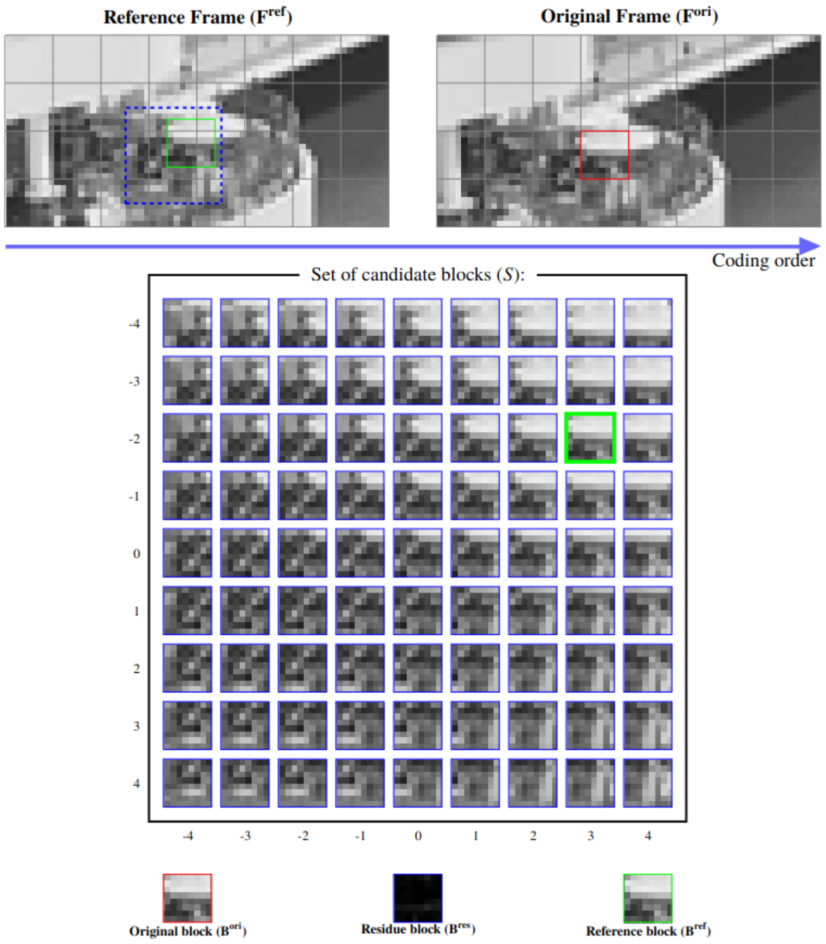


Source: author, BasketballPass frame 58 (BOSSEN; BOYCE, *et al.*, 2020).

frame, or inter prediction, which seeks the reference block in another frame. With this in mind, a list of candidate blocks is created and then searched for a reference block that minimizes a given cost.

For inter prediction, Motion Estimation (ME) is performed. In this step, a reference block is found within a set of candidate blocks, searching for a block with the biggest similarity, this search is known as Block Matching Algorithm (BMA). The residuals are calculated using the reference block, along with its associated MV, so that the decoder can locate the reference block for each residual block and reconstruct the original one. Figure 7 provides an example of the search for the reference block for Integer Motion Estimation (IME). FME and AME are part of the ME step and are used to achieve higher coding gains.

Figure 7 – Example of reference block search.


Reference Frame (F^{ref})
Original Frame (F^{ori})
Coding order
Set of candidate blocks (S):
-4
-3
-2
-1
0
1
2
3
4
-4
-3
-2
-1
0
1
2
3
4
Original block (B^{ori})
Residue block (B^{res})
Reference block (B^{ref})

Source: Seidel (2019).

With a reference block chosen, the residuals are calculated and then subjected to the transform step. The transform process involves converting the residual block data from the spatial domain to the frequency domain. With this transformation, visual changes become concentrated in a few coefficients of the frequency spectrum, which enhances compression in entropy coding (RICHARDSON, 2003).

After the transform step, quantization of the data is performed, a lossy compression step. Quantization of data reduces the range of representation of the signal, introducing losses but allowing for increased compression. The quantization interval is indirectly selected through Quantization Parameter (QP), which is used to control the desired video quality.

2.3 OBJECTIVE METRICS

During video coding, the encoder seeks to minimize a cost that considers both compression, which will be measured using bitrate, and quality, which will be reduced to analyzing the signal distortion, between an original block and a candidate block, a process known as Rate-Distortion Optimization (RDO). The cost used in RDO is a Lagrangian rate-distortion cost (j_{cost}), which weighs the rate and distortion metrics with a Lagrange multiplier (λ), as shown in Equation 4.

$$j_{cost} = rate + \lambda distortion \tag{4}$$

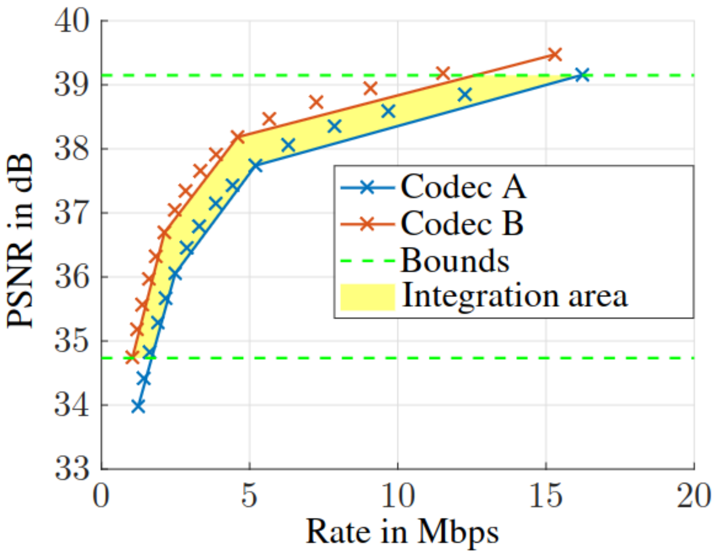
The most widely used distortion metric is Peak Signal-to-Noise Ratio (PSNR) (measured in dB), an objective metric based on mathematical models of signal processing. Its value is calculated using the maximum sample value (peak signal) and the Mean Squared Errors (MSE) of the samples in a block or sequence, as shown in Equation 5.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{5}$$

The BD-Rate metric is commonly used when comparing different video encoders or different configurations of the same encoder. When making a comparison using BD-Rate, the bitrate loss (or gain) for a sequence while maintaining the same objective quality is found (HERGLOTZ; OCH, *et al.*, 2024). The resulting value is the average difference between two Rate-Distortion (RD) curves, expressed as a percentage.

Figure 8 illustrates a comparison between codec A and codec B, where the integration area between their curves represents the BD-Rate value. Notably, for the same distortion (PSNR) value, codec A has a higher rate at each point on the curve, indicating that codec B is the better option.

Figure 8 – Example of RD curves.



Source: Herglotz, Kranzler, *et al.* (2022).

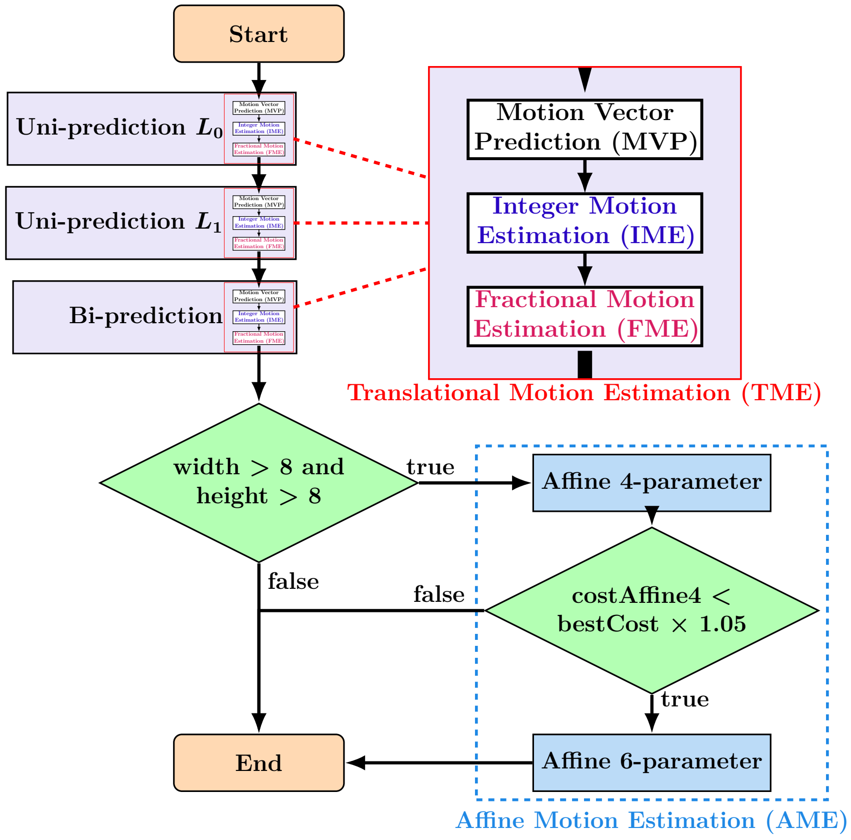
It is also important to note the curve points: those directly on the curve represent results that were encoded, while the curve itself is a simple linear interpolation. Points outside the curve are interpolated using some other method, which usually presents more accurate results. The Akima interpolation method returns more accurate interpolated curves, yielding better results (HERGLOTZ; OCH, *et al.*, 2024). The encoded results used for iterpolation are usually obtained by coding the same sequence with different QP values.

Using BD-Rate as the comparison metric enables a straightforward numerical comparison between different configurations with the same reference, as a higher BD-Rate value stands for a worse overall result.

2.4 FME AND AME IN VVC

Video codecs rely on inter-frame prediction to reduce redundancies between temporally close frames. The two tools focused on this work are part of the Motion Estimation (ME), within VVC inter-frame prediction, illustrated in Figure 9.

Figure 9 – ME steps in a VVC encoder.


Start
Uni-prediction L_0
Uni-prediction L_1
Bi-prediction
Motion Vector Prediction (MVP)
Integer Motion Estimation (IME)
Fractional Motion Estimation (FME)
Translational Motion Estimation (TME)
width > 8 and height > 8
true
false
Affine 4-parameter
false
costAffine4 < bestCost × 1.05
true
Affine 6-parameter
End
Affine Motion Estimation (AME)

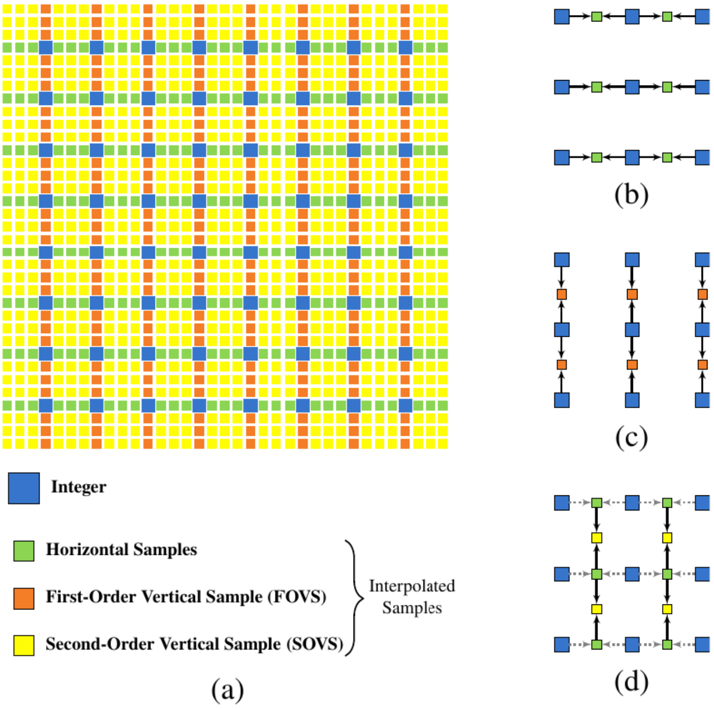
Source: author.

The initial three steps, comprising the uni-directional prediction in the reference frame lists L_0 and L_1 and the bi-directional prediction, are common to previous standards that rely solely on the Translational Motion Estimation (TME). Frame list L_0 is comprised of frames that are temporally **past** the current frame, while list L_1 is made of temporally **future**, bi-directional prediction is performed using the best results from each uni-directional step.

The TME works similarly to how ME was previously explained, being roughly divided into three steps:

1. The Motion Vector Prediction (MVP), required to define a Predicted Motion Vector (PMV), to which the MVs are relative to;
2. The IME, which performs an initial BMA in integer MV positions; and
3. The Fractional Motion Estimation (FME), which refines the search with a new BMA over interpolated candidates at fractional positions (presented in Figure 10), as most of the standards allow for fractional precision MVs.

Figure 10 – Fractional samples for a 8 × 8 block


Integer
Horizontal Samples
First-Order Vertical Sample (FOVS)
Second-Order Vertical Sample (SOVS)
Interpolated Samples
(a)
(b)
(c)
(d)

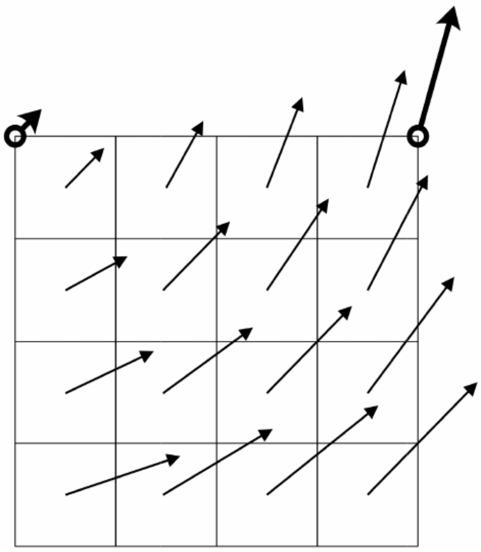
Source: Filho *et al.* (2021).

One of the innovations of VVC is the support of variable precision MVs, a feature called Adaptive Motion Vector Resolution (AMVR) (ZHU *et al.*, 2019; LIU *et al.*, 2019). With AMVR, MVs may be encoded with 1/4-precision, 1/2-precision, integer-precision, and in steps of four samples (BROSS *et al.*, 2021). Also, when using 1/2-precision MVs, the VVC adopts an alternative interpolation filter that improves the coding efficiency. By comparison, in HEVC, all MVs were represented in 1/4-precision (SULLIVAN *et al.*, 2012). Disabling the use of AMVR increases the BD-Rate by an average 1.6%, being a valuable new tool in the standard (CHIEN *et al.*, 2021).

Another innovation of VVC is the AME, which is only performed if the original block size exceeds 8 in both width and height, as shown in Figure 9. Such a requirement is due to the way AME works, as it divides the original block into 4 × 4 sub-blocks.

Instead of coding the MV of each sub-block, a VVC encoder may choose to encode only two or three MVs, called Control Points (CPs). Figure 11 shows an example of affine motion field using two CPs.

Figure 11 – Example of motion field for affine with two CPs.



Source: Yang *et al.* (2021).

The CPs are used to derive the actual MV of each subblock. As there is no associated rate cost to represent these derived MVs, the standard allows them to have 1/16-precision (BROSS *et al.*, 2021). In a way, the TME can be seen as a AME with only one CP.

3 RELATED WORK

This section presents works that analyzed FME and AME, reporting their results and comparing them to this work.

Filho (2022) analyzed the FME in VTM v13.0, and showed that disabling the FME increases the BD-Rate by 0.67% for RA and by 1.21% for LD, much lower than the results reported for HM in (SEIDEL, 2019). These results were unexpected but may be explained by the compensation of other tools, such as the AME.

Yang *et al.* (2021) provide an overview of sub-block motion derivation in VVC, including the AME. The authors report a BD-Rate increase of 3.4% while considering CTC classes A1, A2, B, and C when disabling AME and its refinements in VTM v9.0.

Muñoz *et al.* (2023) also reported an average increase of 2.15% BD-Rate for RA when disabling the AME in VTM v9.0.

The information above is presented in Table 2 for an easier comparison between findings, reporting the used VTM version along with BD-Rate data.

Table 2 – Related works BD-Rate for FME and AME.

Source	**VTM Version**	**FME Disabled**	**AME Disabled**	**Both Disabled**
Filho (2022)	13.0	0.67–1.21%	-	-
Yang *et al.* (2021)	9.0	-	3.4%	-
Muñoz *et al.* (2023)	9.0	-	2.15%	-
This work	23.1	0.63–0.93%	2.45–3.73%	4.12–5.05%

Source: author.

The cited related work presents analysis only on one portion of ME at a time, either the FME or AME, but not both simultaneously, nor the effect they have on each other. This work analyzes both tools working concurrently, allowing for results regarding the effect they have on each other.

4 METHOD

This chapter presents the steps taken to reach the results found in the next chapter, presenting the changes made to the reference software VTM, along with what was used for the analysis and how the analysis itself was made.

The reference software from a standard is the software used when investigating tools, benchmarking or testing in general, serving as a baseline for research. To analyze the interaction between FME and AME, four distinct experiments were planned using the VTM v23.1 (BOSSEN; LI; SUEHRING, 2024), by either enabling or disabling the execution of FME and AME, together or separately, as listed in Table 3.

To disable AME, the Command Line Interface (CLI) option `-aff 0` was used. However, disabling FME required modifications to the VTM source code, as no option was readily available via CLI. In this case, a new option named `-fme` was created by modifying the VTM source code.

Table 3 – Coding configurations.

Configuration	**FME**	**AME**	**CLI**
Baseline	Enabled	Enabled	`-fme 1 -aff 1`
FME Disabled	Disabled	Enabled	`-fme 0 -aff 1`
AME Disabled	Enabled	Disabled	`-fme 1 -aff 0`
Both Disabled	Disabled	Disabled	`-fme 0 -aff 0`

Source: author.

Reference software is usually associated with guidelines for testing, known as Common Test Conditions (CTC), defining which sequences should be tested and the reference software configuration that should be used during coding. For every coding configuration, each of the CTC sequences was encoded in both RA and LD temporal configurations, using QPs 22, 27, 32, and 37, totaling 32 codings per sequence.

Sequences in classes A1 and A2 are encoded using only the RA temporal configuration as per CTC (BOSSEN; BOYCE, *et al.*, 2020) recommendation. Additionally, for sequences in these classes, only the first second of each sequence was encoded. Table 4 shows each encoded sequence, along with class, resolution, and how many frames were encoded. Classes group sequences with similar characteristics, classes A to F cover resolutions from Ultra High-Definition (UHD) (3840×2160) to Wide Quarter Video Graphics Array (WQVGA) (416×240), with sequences from classes A-D consisting of natural captured camera content, class E being live feeds, and class F containing computer-generated content (CHIEN *et al.*, 2021).

From the coding results, reported PSNR and bitrate data were collected and then used to calculate the BD-Rate of each configuration relative to the Baseline (FME and AME both enabled). To obtain the sequences' BD-Rate, the `bjontegaard` Python package (HERGLOTZ, CHRISTIAN, 2024) was used, configured with the Akima interpolation

Table 4 – Tested sequences list and their characteristics.

Class	Sequence	Resolution	FPS	Encoded Frames
A1	Tango2	3840 × 2160	60	60
	FoodMarket4	3840 × 2160	60	60
	Campfire	3840 × 2160	30	30
A2	CatRobot1	3840 × 2160	60	60
	DaylightRoad2	3840 × 2160	60	60
	ParkRunning3	3840 × 2160	50	50
B	MarketPlace	1920 × 1080	60	600
	RitualDance	1920 × 1080	60	600
	Cactus	1920 × 1080	50	500
	BasketballDrive	1920 × 1080	50	500
	BQTerrace	1920 × 1080	60	600
C	RaceHorsesC	832 × 480	30	300
	BQMall	832 × 480	60	600
	PartyScene	832 × 480	50	500
	BasketballDrill	832 × 480	50	500
D	RaceHorses	416 × 240	30	300
	BQSquare	416 × 240	60	600
	BlowingBubbles	416 × 240	50	500
	BasketballPass	416 × 240	50	500
E	FourPeople	1280 × 720	60	600
	Johnny	1280 × 720	60	600
	KristenAndSara	1280 × 720	60	600
F	BasketballDrillText	832 × 480	50	500
	ChinaSpeed	1024 × 768	50	500
	ArenaOfValor	1920 × 1080	60	600
	SlideEditing	1280 × 720	30	300
	SlideShow	1280 × 720	50	500

Source: author.

method, since it yields more accurate interpolation results, as recommended by (HERGLOTZ; OCH, *et al.*, 2024).

5 RESULTS

Tables 5 and 6 present the BD-Rate results for RA and LD, respectively. In both tables, the results are shown individually for each tested sequence. In addition to the results for the coding configurations in Table 3, is also included in the presented results is the ***sum*** of the BD-Rates of FME Disabled and AME Disabled configurations to contrast it with the configuration where both tools are disabled.

Table 5 – RA temporal configuration BD-Rate for each sequence.

Sequence	**FME Disabled**	**AME Disabled**	**Both Disabled**	***Sum****
Tango2	0.099	1.372	1.524	1.471
FoodMarket4	-0.046	0.314	0.410	0.268
Campfire	0.198	0.181	0.453	0.379
A1 Average	0.084	0.622	0.796	0.706
CatRobot1	0.293	7.168	8.595	7.460
DaylightRoad2	0.698	6.550	9.158	7.247
ParkRunning3	0.394	3.947	4.619	4.341
A2 Average	0.462	5.888	7.457	6.349
MarketPlace	0.647	4.040	5.690	4.687
RitualDance	0.521	2.249	3.623	2.770
Cactus	0.258	7.440	8.826	7.697
BasketballDrive	0.911	1.258	2.891	2.168
BQTerrace	1.026	0.184	2.935	1.209
B Average	0.673	3.034	4.793	3.706
RaceHorsesC	1.383	1.441	4.089	2.824
BQMall	0.939	0.962	2.789	1.900
PartyScene	1.117	1.882	4.755	2.999
BasketballDrill	0.530	0.593	2.402	1.123
C Average	0.992	1.220	3.509	2.212
RaceHorses	1.779	1.248	4.232	3.027
BQSquare	0.941	1.465	4.692	2.406
BlowingBubbles	1.537	1.112	3.924	2.649
BasketballPass	1.318	0.549	2.598	1.867
D Average	1.394	1.094	3.861	2.487
FourPeople	0.310	1.068	1.803	1.378
Johnny	0.393	1.785	2.965	2.178
KristenAndSara	0.379	1.849	2.764	2.229
E Average	0.361	1.567	2.511	1.928
BasketballDrillText	0.566	0.451	2.480	1.017
ChinaSpeed	0.129	2.743	3.069	2.872
ArenaOfValor	0.510	1.045	2.016	1.554
SlideEditing	0.001	-0.059	-0.016	-0.059
SlideShow	0.234	14.146	17.922	14.380
F Average	0.288	3.665	5.094	3.953
Average	0.632	2.481	4.119	3.113

****Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)
Source: author.

Figures 12 and 13 present the same data as the tables in a visual way. Aiming to improve visualization of the overall order between experiments, results are connected using a dashed line. The outlier values from SlideShow in Figure 12 were removed to further enhance the visualization. These results are discussed in more detail when analyzing Figure

Table 6 – LD temporal configuration BD-Rate for each sequence.

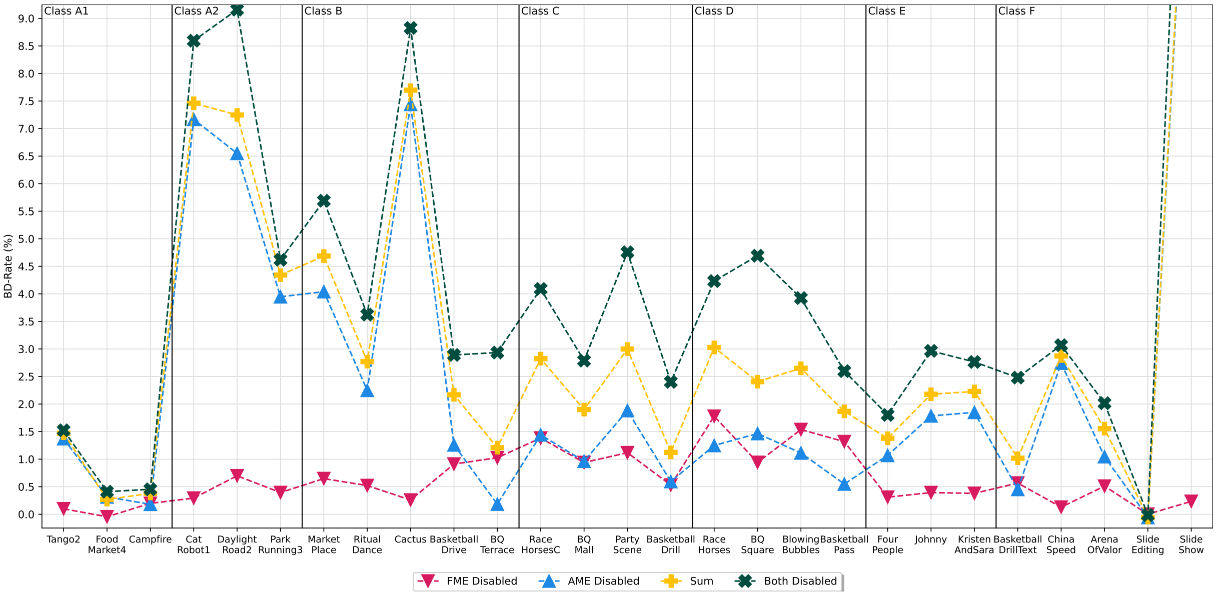
Sequence	**FME Disabled**	**AME Disabled**	**Both Disabled**	***Sum***[*]
MarketPlace	0.768	3.483	4.980	4.250
RitualDance	0.541	1.976	2.348	2.517
Cactus	0.399	10.064	10.860	10.463
BasketballDrive	1.159	1.885	3.357	3.043
BQTerrace	0.543	1.143	-1.195	1.686
B Average	0.682	3.710	4.070	4.392
RaceHorsesC	0.903	1.432	2.885	2.335
BQMall	1.010	1.701	3.776	2.711
PartyScene	1.619	5.175	8.588	6.794
BasketballDrill	0.941	0.982	2.846	1.922
C Average	1.118	2.322	4.524	3.441
RaceHorses	0.831	1.666	2.871	2.496
BQSquare	2.224	13.323	18.385	15.547
BlowingBubbles	2.109	3.615	7.239	5.724
BasketballPass	1.263	0.962	2.578	2.224
D Average	1.607	4.891	7.768	6.498
FourPeople	0.811	2.065	3.226	2.876
Johnny	0.649	5.064	6.340	5.712
KristenAndSara	0.686	4.194	5.141	4.881
E Average	0.715	3.774	4.902	4.490
BasketballDrillText	0.847	0.927	2.419	1.775
ChinaSpeed	0.560	2.175	2.024	2.735
ArenaOfValor	0.515	1.635	1.883	2.150
SlideEditing	0.347	0.259	0.463	0.606
SlideShow	0.851	14.593	15.055	15.445
F Average	0.624	3.918	4.369	4.542
Average	0.932	3.729	5.051	4.662

[*] ***Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)

Source: author.

14, while the actual values are presented in Tables 5 and 6.

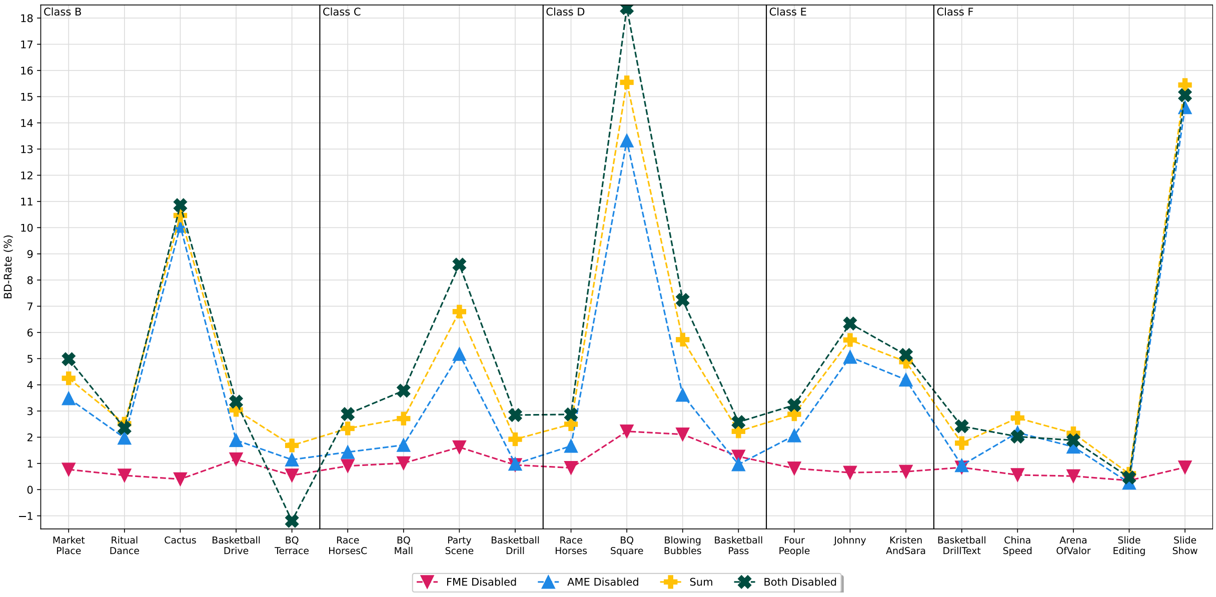
Figure 12 – RA temporal configuration BD-Rate for each sequence.



Source: author.

Notice that the y-axis in Figure 12 represents a smaller BD-Rate interval than in Figure 13.

Figure 13 – LD temporal configuration BD-Rate for each sequence.



Source: author.

Thus, in most cases, disabling a tool in RA has a lower impact than disabling it in LD. However, this largely depends on the sequence content. For instance, taking only the results of **Both Disabled** into consideration, the BD-Rate difference between RA and LD for the Cactus sequence is relatively small, 8.83% (RA) and 10.86% (LD), while there is a major difference for the BQSquare sequence, 4.69% (RA) and 18.38% (LD).

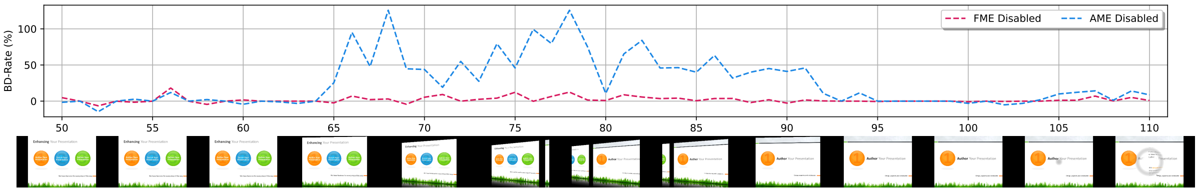
Nonetheless, for most sequences, in both RA and LD, AME has a larger impact than the FME, as can be noticed by the higher values of BD-Rate for **AME Disabled** than for **FME Disabled**. This result was expected, as being able to capture rotation, shearing, and zooming, the AME tool is versatile, bringing a significant enhancement in the coding efficiency of the majority of tested sequences.

As seen in both figures and tables, AME particularly enhances sequences:

- From class A2: CatRobot1 has an object that rotates, while DaylightRoad2 and ParkRunning3 have zooming;
- Cactus: similarly to CatRobot1, it has a rotating object;
- BQSquare: the camera is zooming and moving;
- SlideShow: the slide transitions present in this sequence are very characteristic affine transformations, such as shearing and zooming.

To show in greater detail the results of **FME Disabled** and **AME Disabled** for the SlideShow sequence in RA configuration, Figure 14 presents the BD-Rate computed for each frame of the sequence, along with a miniature of one in every 10 frames to provide a clue of the video content and how it relates to the BD-Rate. One may notice that when the sequence is around frame 65, the BD-Rate of **AME Disabled** starts to rise.

Figure 14 – BD-Rate per frame for SlideShow sequence considering RA configuration, with the horizontal axis showing the frame index.



Source: author.

This corresponds to the beginning of a slide transition that ends around frame 95, when the BD-Rate of **AME Disabled** goes back to near zero. The transition in question is a prime example of an affine transformation. Thus, without the AME tool, the encoder expends a large amount of the available rate to signalize MVs, resulting in a sequence BD-Rate of 17.92%. Furthermore, the **FME Disabled** results in Figure 14 are almost negligible, as can also be noticed by the resulting sequence BD-Rate in Tables 5 (0.234%) and 6 (0.851%).

In fact, analyzing the overall **FME Disabled** results shows that the FME has negligible impacts when AME is enabled. Nevertheless, for low-resolution sequences, such as the ones from class D, the FME still has its importance. The effect can be more easily observed in Table 7, which presents the average increase in BD-Rate per class for each configuration tested.

Table 7 – Average BD-Rate for each class.

Cfg.	Class	FME Disabled	AME Disabled	*Sum**	Both Disabled
Random Access (RA)	A1	0.084	0.622	0.706	0.796
	A2	0.462	5.888	6.349	7.457
	B	0.673	3.034	3.706	4.793
	C	0.992	1.220	2.212	3.509
	D	1.394	1.094	2.487	3.861
	E	0.361	1.567	1.928	2.511
	F	0.288	3.665	3.953	5.094
	Average	0.632	2.481	3.113	4.119
Low Delay (LD)	B	0.682	3.710	4.392	4.070
	C	1.118	2.322	3.441	4.524
	D	1.607	4.891	6.498	7.768
	E	0.715	3.774	4.490	4.902
	F	0.624	3.918	4.542	4.369
	Average	0.932	3.729	4.662	5.051

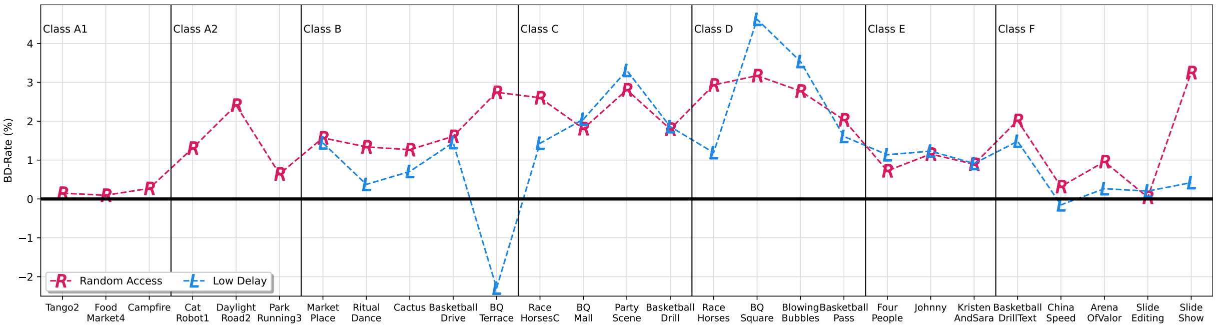
****Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)
Source: author.

When only the FME is disabled, there is an average increase of 0.63% and 0.93% BD-Rate for RA and LD. The increase is bigger for classes C and D in both cases, corresponding to lower-resolution sequences. With a few exceptions in LD, the most striking being BQTerace, **Both Disabled** results in the highest BD-Rate values, which is expected. However, it is worth comparing the BD-Rate results from **Sum** with **Both Disabled**. Notice that in most cases, the **Sum** of BD-Rates is smaller than the BD-Rate of **Both Disabled**.

Considering the ME flow described in Figure 9 and this result, it is possible to conclude that AME is compensating for the lack of FME to some degree, which is in line with the hypothesis previously defined. On the other hand, even in light of this AME compensation, the impact of fully disabling the FME is still short of what was observed in previous standards and thus should be further studied.

Another way of analyzing the FME effect is through Figure 15. The figure presents the BD-Rate for the sequences tested in both RA and LD configurations, but instead of using the usual reference of **Both Enabled**, **AME Disabled** is used as reference and compared to **Both Disabled**, *i.e.*, FME gains in a scenario where AME is not available.

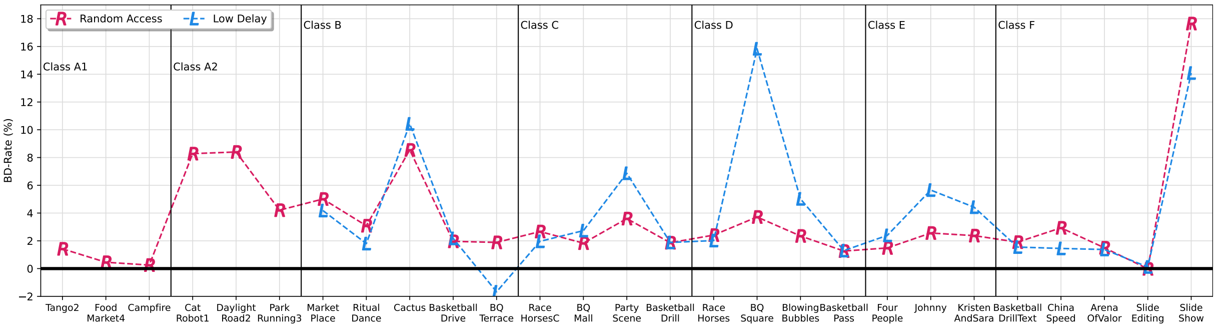
Figure 15 – **Both Disabled** BD-Rate using **AME Disabled** as reference.



Source: author.

In a similar fashion, Figure 16 shows the BD-Rate for AME using the same technique: using **FME Disabled** as the reference and comparing it to **Both Disabled**. Note the difference in y-axis scale for both cases, demonstrating a greater impact of AME in general.

Figure 16 – **Both Disabled** BD-Rate using **FME Disabled** as reference.



Source: author.

Figures 12, 13, 15, and 16 present similar results, highlighting that even though the data is analyzed from a different point of view, the results found and analysis made so far are reasonable.

6 CONCLUSION

This work analyzed the coding efficiency of the Fractional Motion Estimation (FME) and Affine Motion Estimation (AME) within the VTM encoder, which is the reference implementation of the VVC standard. Firstly, our results show that disabling the FME in VTM causes a significantly smaller increase in BD-Rate relative to doing the same thing in the HM, the reference encoder of VVC predecessor, the HEVC, being reduced from 12% to less than 1%.

Also, the newly introduced AME tool enhances the coding efficiency of VTM by an average of 2.48% BD-Rate (RA) and 3.73% (LD). These results are compatible with the ones from the presented literature. Sequences that contain characteristic affine transformations have significant gains from AME, as shown by the entirety of class A2 and especially SlideShow.

By analyzing VVC ME flow and the results obtained from disabling both tools, it is also possible to conclude that AME can compensate for the loss of FME to some degree in the majority of tested sequences. Such a conclusion relies upon the fact that the BD-Rates from disabling both tools are larger than the sum of the BD-Rates of disabling them individually. Then again, the observed impact of disabling the FME still falls short of the expected.

This work sheds light on how tools interact and overlap during video coding, particularly in the ME step. Both FME and AME were analyzed, and data from multiple encodings were used to assess their impact on the coding efficiency of the VVC reference software. Furthermore, it was shown that analyzing tools working together provides valuable insights, revealing results that would not be possible by studying them in isolation.

Future work includes analyzing whether the 6-parameter AME is executed more times when skipping FME due to worse RD costs from IME alone and analyzing the effect of AMVR, so as to verify if the VTM is capable of noticing that no fractional MV is being found by TME.

REFERENCES

BOSSEN, F.; BOYCE, J., *et al.* **VTM common test conditions and software reference configurations for SDR video**. 2020. Disponível em: `https://jvet-experts.org/doc_end_user/current_document.php?id=10545`.

BOSSEN, Frank; LI, Xiang; SUEHRING, Karsten. **VVC Test Model (VTM) v23.1**. 2024. Disponível em: `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-23.1`.

BROSS, Benjamin *et al.* Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 31, n. 10, p. 3736–3764, Oct. 2021. ISSN 1558-2205. DOI: `10.1109/tcsvt.2021.3101953`.

CHIEN, Wei-Jung *et al.* Motion Vector Coding and Block Merging in the Versatile Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 31, n. 10, p. 3848–3861, Oct. 2021. ISSN 1558-2205. DOI: `10.1109/tcsvt.2021.3101212`.

FILHO, Vanio Rodrigues. **Fixed search patterns and VLSI architecture for the efficient computation of the versatile video coding fractional motion estimation**. 2022. PhD thesis – Universidade Federal de Santa Catarina. Disponível em: `https://repositorio.ufsc.br/handle/123456789/234693`.

FILHO, Vanio Rodrigues *et al.* Hardware-Friendly Search Patterns for the Versatile Video Coding Fractional Motion Estimation. IEEE, Oct. 2021. DOI: `10.1109/mmsp53017.2021.9733603`.

HERGLOTZ, Christian; KRANZLER, Matthias, *et al.* Beyond Bjøntegaard: Limits of Video Compression Performance Comparisons. IEEE, Oct. 2022. DOI: `10.1109/icip46576.2022.9897912`.

HERGLOTZ, Christian; OCH, Hannah, *et al.* The Bjøntegaard Bible Why Your Way of Comparing Video Codecs May Be Wrong. **IEEE Transactions on Image Processing**, Institute of Electrical and Electronics Engineers (IEEE), v. 33, p. 987–1001, 2024. ISSN 1941-0042. DOI: `10.1109/tip.2023.3346695`.

HERGLOTZ, CHRISTIAN. **bjontegaard v1.3.0**. 2024. Disponível em: `https://pypi.org/project/bjontegaard/`.

HUANG, Yu-Wen *et al.* Block Partitioning Structure in the VVC Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 31, n. 10, p. 3818–3833, Oct. 2021. ISSN 1558-2205. DOI: `10.1109/tcsvt.2021.3088134`.

HUNT, R W G. **The reproduction of colour**. 6. ed. Nashville, TN: John Wiley & Sons, May 2005. (The Wiley-IS&T Series in Imaging Science and Technology).

ISO CENTRAL SECRETARY. **Information technology – Coded representation of immersive media – Part 3: Versatile video coding**. Geneva, CH, 2020.

________. **Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding**. Geneva, CH, 2013.

LIU, Hongbin *et al.* **JVET-M0255: AHG11: MMVD without Fractional Distances for SCC**. Marrakech, MA, Jan. 2019.

MUÑOZ, Marcello M. *et al.* Hardware Design for the Affine Motion Compensation of the VVC Standard. IEEE, p. 1–4, Feb. 2023. DOI: `10.1109/lascas56464.2023.10108350`.

RICHARDSON, Iain E. **H.264 and MPEG-4 video compression**. Chichester, England: John Wiley & Sons, Oct. 2003.

SANDVINE. **Global Internet Phenomena Report 2024**. 2024. Disponível em: `https://www.sandvine.com/phenomena`.

SEIDEL, Ismael. **ANÁLISE DO IMPACTO DE PEL DECIMATION NA CODIFICAÇÃO DE VÍDEOS DE ALTA RESOLUÇÃO**. 2014. MA thesis – Universidade Federal de Santa Catarina. Disponível em: `https://repositorio.ufsc.br/handle/123456789/129309`.

________. **Exploiting SATD properties to reduce energy in video coding**. 2019. PhD thesis – Universidade Federal de Santa Catarina. Disponível em: `https://repositorio.ufsc.br/handle/123456789/216224`.

SHI, Yun-Qing; SUN, Huifang. **Image and video compression for multimedia engineering**. 3. ed. London, England: CRC Press, Mar. 2019. (Image Processing Series).

SIQUEIRA, Icaro; CORREA, Guilherme; GRELLERT, Mateus. Complexity and Coding Efficiency Assessment of the Versatile Video Coding Standard. IEEE, v. 9217, p. 1–5, May 2021. DOI: `10.1109/iscas51556.2021.9401714`.

SULLIVAN, G.J. *et al.* Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 12, p. 1649–1668, Dec. 2012.

YANG, Haitao *et al.* Subblock-Based Motion Derivation and Inter Prediction Refinement in the Versatile Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, Institute of Electrical and Electronics Engineers

(IEEE), v. 31, n. 10, p. 3862–3877, Oct. 2021. ISSN 1558-2205. DOI: 10.1109/tcsvt.2021.3100744.

ZHU, Weijia *et al.* **JVET-N0260-v1: Non-CE8: Adaptive Fractional MVD search in DMVR for SCC**. Geneva, CH, Mar. 2019.

Appendix

Coding Efficiency Analysis of Fractional Motion Estimation and Affine Motion Estimation in VVC

João Mai[1]

[1]Departamento de Informática e Estatıstica – Centro Tecnológico
Universidade Federal de Santa Catarina (UFSC)

Abstract. *The VVC standard emerged to circumvent the ever-higher demands imposed by high-resolution video consumption. VVC improves upon well-established tools, such as the FME, as well as brings new ones, such as the AME, increasing the coding efficiency when compared to its predecessor, the HEVC. Related work demonstrated that the overall coding efficiency of the FME was reduced in the VTM compared to HM. Thus, it is hypothetised that the AME may be interacting with FME once AME also relies on the generation of interpolated samples at possible fractional positions. Therefore, this work analyzes the coding efficiency of both tools, FME and AME, considering their implementation in the VTM. This work shows that disabling the AME reduces the average coding efficiency by 2.48% (RA) and 3.73% (LD) while disabling the FME reduces, on average, 0.63% (RA) and 0.93% (LD), confirming the lower impact of FME. When disabling both tools, the average coding efficiency is reduced by 4.19% (RA) and 5.05% (LD), which are higher than the sum of the individual coding efficiency losses, confirming a small but present AME compensatory effect.*

Resumo. *O padrão VVC surgiu para contornar as exigências cada vez maiores impostas pelo consumo de vídeo em alta resolução. O VVC melhora ferramentas bem estabelecidas, como a FME, além de introduzir novas, como a AME, aumentando a eficiência de codificação em comparação ao seu antecessor, o HEVC. Trabalhos relacionados demonstraram que a eficiência de codificação da FME foi reduzida no VTM em comparação ao HM. Assim, é hipotetizado que a AME possa estar interagindo com a FME, uma vez que a AME também depende da geração de amostras interpoladas em possíveis posições fracionárias. Portanto, este trabalho analisa a eficiência de codificação de ambas as ferramentas, FME e AME, considerando suas implementações no VTM. Também mostra-se que desabilitar a AME reduz a eficiência média de codificação em 2,48% (RA) e 3,73% (LD), enquanto desabilitar a FME reduz em média 0,63% (RA) e 0,93% (LD), confirmando o menor impacto da FME. Ao desabilitar ambas as ferramentas, a eficiência média de codificação é reduzida em 4,19% (RA) e 5,05% (LD), valores superiores à soma das perdas individuais de eficiência de codificação, confirmando um pequeno, mas presente, efeito compensatório da AME.*

1. Introduction

The last decade saw substantial improvements in Internet connections worldwide, which, paired with the popularity of smartphones and other portable devices, resulted in a significant increase in digital video consumption. In 2023, 54% of data volume on non-portable devices (such as computers and smart TVs) corresponded to on-demand streaming (platforms like YouTube and Netflix), with 57% for the same category on mobile devices [Sandvine 2024].

In the face of a demand for 4K (3840×2160 pixels per frame) and higher resolutions, more efficient compression standards are made necessary, employing new tools and methods capable of enabling the transmission and storage of videos at those higher resolutions. In this context, the Joint Video Experts Team (JVET) introduced the Versatile Video Coding (VVC) standard [ISO Central Secretary 2020] in 2020, succeeding the High Efficiency Video Coding (HEVC) [ISO Central Secretary 2013].

When comparing the reference software of the two standards, VVC Test Model (VTM) and HEVC Test Model (HM), the former brings an average 38.9% Bjøntegaard Delta Rate (BD-Rate) improvement in coding efficiency for Random Access (RA) configuration and 30.9% for Low Delay (LD) configuration, however, with coding time increases of 803% and 659%, respectively [Bross et al. 2021]. The increase in coding efficiency and coding time in VTM is attributed to the higher complexity of VVC, which not only enhances well-established tools, such as the Fractional Motion Estimation (FME), but also adopts new ones, such as the Affine Motion Estimation (AME) [Bross et al. 2021, Yang et al. 2021].

While the FME tool in the HM was one of the culprits for inter-prediction high complexity, the FME in VTM corresponds to a smaller percentage of the overall runtime, with the average measurement being 60% of its predecessor [Siqueira et al. 2021], which is partially explained by Single Instruction Multiple Data (SIMD) optimizations. Moreover, fully disabling the FME in HM degrades the coding efficiency by 10.89% and 12.75%, average BD-Rate, for RA and LD configurations, respectively [Seidel 2019]. Yet, the cost of disabling the FME in VTM is smaller, with average BD-Rate of 0.67% (RA) and 1.21% (Low Delay with P-slices only (LDP)) [Filho et al. 2021].

The adoption of AME increases the coding time of VTM by 27%, on average [Yang et al. 2021], reducing the rate for Motion Vector (MV) signaling and thus improving coding efficiency by 3.4%. In fact, given that AME produces a block-based motion field with 1/16-precision MVs, the hypothesis is that the AME duplicates certain aspects of the FME, thus partially justifying the lower coding efficiency impact of FME in VTM.

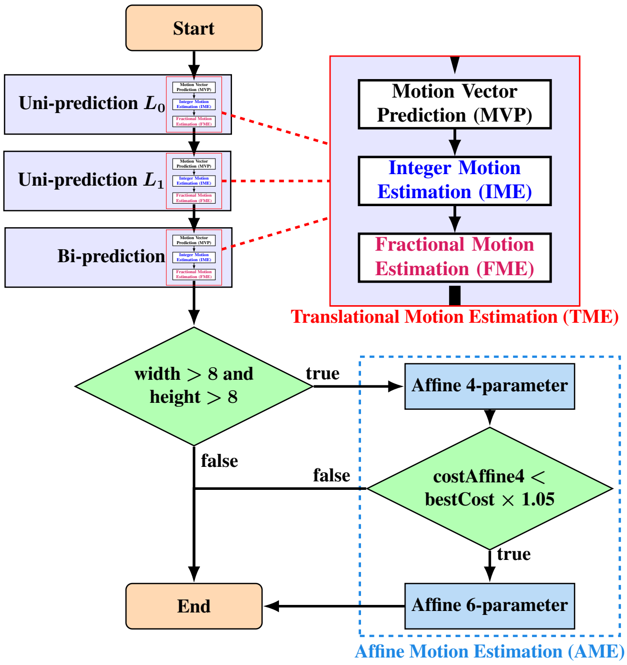
This work aims to provide an overview in regards to the inter-prediction step in VVC, analyzing the FME and AME tools impact on coding efficiency. The goal is to better understand the trade-off between compression and complexity associated with each tool, shedding light to the conundrum that is introducing higher complexity tools without understanding how they interfere with each other, as most works tend to focus on tools in isolation.

2. Background

Video codecs rely on inter-frame prediction to reduce redundancies between temporally close frames. The two tools focused on this work are part of the Motion Estimation (ME), within VVC inter-frame prediction, illustrated in Figure 1.

The initial three steps, comprising the uni-directional prediction in the reference frame lists L_0 and L_1 and the bi-directional prediction, are common to previous stan-

Figure 1. Motion Estimation (ME) steps in a VVC encoder.


Start
Uni-prediction L_0
Uni-prediction L_1
Bi-prediction
Motion Vector Prediction (MVP)
Integer Motion Estimation (IME)
Fractional Motion Estimation (FME)
Translational Motion Estimation (TME)
width > 8 and height > 8
true
false
Affine 4-parameter
costAffine4 < bestCost × 1.05
false
true
Affine 6-parameter
End
Affine Motion Estimation (AME)

dards that rely solely on the Translational Motion Estimation (TME). Frame list L_0 is comprised of frames that are temporally **past** the current frame, while list L_1 is made of temporally **future**, bi-directional prediction is performed using the best results from each uni-directional step.

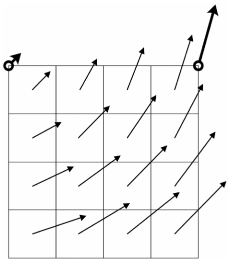
The TME can be roughly divided into three stages:

1. The Motion Vector Prediction (MVP), required to define a Predicted Motion Vector (PMV), to which the MVs are relative to;
2. The Integer Motion Estimation (IME), which performs an initial Block Matching Algorithm (BMA) in integer MV positions; and
3. The Fractional Motion Estimation (FME), which refines the search with a new BMA over interpolated candidates at fractional positions (presented in Figure **??**), as most of the standards allow for fractional precision MVs.

One of the innovations of VVC is the support of variable precision MVs, a feature called Adaptive Motion Vector Resolution (AMVR) [Zhu et al. 2019, Liu et al. 2019]. With AMVR, MVs may be encoded with 1/4-precision, 1/2-precision, integer-precision, and in steps of four samples [Bross et al. 2021]. Also, when using 1/2-precision MVs, the VVC adopts an alternative interpolation filter that improves the coding efficiency. By comparison, in HEVC, all MVs were represented in 1/4-precision [Sullivan et al. 2012]. Disabling the use of AMVR increases the BD-Rate by an average 1.6%, being a valuable new tool in the standard [Chien et al. 2021].

Another innovation of VVC is the AME, which is only performed if the original block size exceeds 8 in both width and height, as shown in Figure 1. Such a requirement is due to the way AME works, as it divides the original block into 4×4 sub-blocks. Instead of coding the MV of each sub-block, a VVC encoder may choose to encode only two or three MVs, called Control Points (CPs). Figure 2 shows an example of affine motion field using two CPs. As there is no associated rate cost to represent these derived MVs, the

Figure 2. Example of motion field for affine with two CPs.



standard allows them to have 1/16-precision [Bross et al. 2021]. In a way, the TME can be seen as a AME with only one CP.

3. Related Works

[Filho 2022] analyzed the FME in VTM v13.0, and showed that disabling the FME increases the BD-Rate by 0.67% for RA and by 1.21% for LD, much lower than the results reported for HM in [Seidel 2019]. These results were unexpected but may be explained by the compensation of other tools, such as the AME. [Yang et al. 2021] provide an overview of sub-block motion derivation in VVC, including the AME. The authors report a BD-Rate increase of 3.4% while considering Common Test Conditions (CTC) classes A1, A2, B, and C when disabling AME and its refinements in VTM v9.0. [Muñoz et al. 2023] also reported an average increase of 2.15% BD-Rate for RA when disabling the AME in VTM v9.0. The information above is presented in the Table 1 for an easier comparison between findings, reporting the used VTM version along with BD-Rate data.

Table 1. Related works BD-Rate for FME and AME.

Source	VTM Version	FME Disabled	AME Disabled	Both Disabled
[Filho 2022]	13.0	0.67–1.21%	-	-
[Yang et al. 2021]	9.0	-	3.4%	-
[Muñoz et al. 2023]	9.0	-	2.15%	-
This work	23.1	0.63–0.93%	2.45–3.73%	4.12–5.05%

The cited related work presents analysis only on one portion of ME at a time, either the FME or AME, but not both simultaneously, nor the effect they have on each other. This work analyzes both tools working concurrently, allowing for results regarding the effect they have on each other.

4. Method

The reference software from a standard is used when investigating tools, benchmarking or testing in general, serving as a baseline for research. To analyze the interaction between FME and AME, four distinct experiments were planned using the VTM v23.1 [Bossen et al. 2024], by either enabling or disabling the execution of FME and AME, together or separately, as listed in Table 2.

To disable AME, the Command Line Interface (CLI) option `-aff 0` was used. However, disabling FME required modifications to the VTM source code, as no option was readily available via CLI. In this case, a new option named `-fme` was created by modifying the VTM source code.

Table 2. Coding configurations.

Configuration	FME	AME	CLI
Baseline	Enabled	Enabled	`-fme 1 -aff 1`
FME Disabled	Disabled	Enabled	`-fme 0 -aff 1`
AME Disabled	Enabled	Disabled	`-fme 1 -aff 0`
Both Disabled	Disabled	Disabled	`-fme 0 -aff 0`

This work follows guidelines for testing, known as Common Test Conditions (CTC), which defines how sequences should be tested and the reference software configuration that should be used during coding. For every coding configuration, each of the CTC sequences was encoded in both RA and LD temporal configurations, using Quantization Parameters (QPs) 22, 27, 32, and 37, totaling 32 codings per sequence.

Sequences in classes A1 and A2 are encoded using only the RA temporal configuration as per CTC [Bossen et al. 2020] recommendation. Additionally, for sequences in these classes, only the first second of each sequence was encoded. Table **??** shows each encoded sequence, along with class, resolution, and how many frames were encoded. Classes group sequences with similar characteristics, classes A to F cover resolutions from Ultra High-Definition (UHD) (3840×2160) to Wide Quarter Video Graphics Array (WQVGA) (416×240), with sequences from classes A-D consisting of natural captured camera content, class E being live feeds, and class F containing computer-generated content [Chien et al. 2021].

From the coding results, reported Peak Signal-to-Noise Ratio (PSNR) and bitrate data were collected and then used to calculate the BD-Rate of each configuration relative to the Baseline (FME and AME both enabled). To obtain the sequences' BD-Rate, the `bjontegaard` Python package [Herglotz, Christian 2024] was used, configured with the Akima interpolation method, since it yields more accurate interpolation results, as recommended by [Herglotz et al. 2024].

5. Results

Tables 3 and 4 present the BD-Rate results for RA and LD, respectively. In both tables, the results are shown individually for each tested sequence. In addition to the results for the coding configurations in Table 2, is also included in the presented results is the ***sum*** of the BD-Rates of FME Disabled and AME Disabled configurations to contrast it with the configuration where both tools are disabled.

Figures 3 and 4 present the same data as the tables in a visual way. Aiming to improve visualization of the overall order between experiments, results are connected using a dashed line. The outlier values from SlideShow in Figure 3 were removed to further enhance the visualization. These results are discussed in more detail when analyzing Figure 5, while the actual values are presented in Tables 3 and 4.

Table 3. RA temporal configuration BD-Rate for each sequence.

Sequence	FME Disabled	AME Disabled	Both Disabled	*Sum**
Tango2	0.099	1.372	1.524	1.471
FoodMarket4	-0.046	0.314	0.410	0.268
Campfire	0.198	0.181	0.453	0.379
A1 Average	0.084	0.622	0.796	0.706
CatRobot1	0.293	7.168	8.595	7.460
DaylightRoad2	0.698	6.550	9.158	7.247
ParkRunning3	0.394	3.947	4.619	4.341
A2 Average	0.462	5.888	7.457	6.349
MarketPlace	0.647	4.040	5.690	4.687
RitualDance	0.521	2.249	3.623	2.770
Cactus	0.258	7.440	8.826	7.697
BasketballDrive	0.911	1.258	2.891	2.168
BQTerrace	1.026	0.184	2.935	1.209
B Average	0.673	3.034	4.793	3.706
RaceHorsesC	1.383	1.441	4.089	2.824
BQMall	0.939	0.962	2.789	1.900
PartyScene	1.117	1.882	4.755	2.999
BasketballDrill	0.530	0.593	2.402	1.123
C Average	0.992	1.220	3.509	2.212
RaceHorses	1.779	1.248	4.232	3.027
BQSquare	0.941	1.465	4.692	2.406
BlowingBubbles	1.537	1.112	3.924	2.649
BasketballPass	1.318	0.549	2.598	1.867
D Average	1.394	1.094	3.861	2.487
FourPeople	0.310	1.068	1.803	1.378
Johnny	0.393	1.785	2.965	2.178
KristenAndSara	0.379	1.849	2.764	2.229
E Average	0.361	1.567	2.511	1.928
BasketballDrillText	0.566	0.451	2.480	1.017
ChinaSpeed	0.129	2.743	3.069	2.872
ArenaOfValor	0.510	1.045	2.016	1.554
SlideEditing	0.001	-0.059	-0.016	-0.059
SlideShow	0.234	14.146	17.922	14.380
F Average	0.288	3.665	5.094	3.953
Average	0.632	2.481	4.119	3.113

****Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)

Figure 3. RA temporal configuration BD-Rate for each sequence.

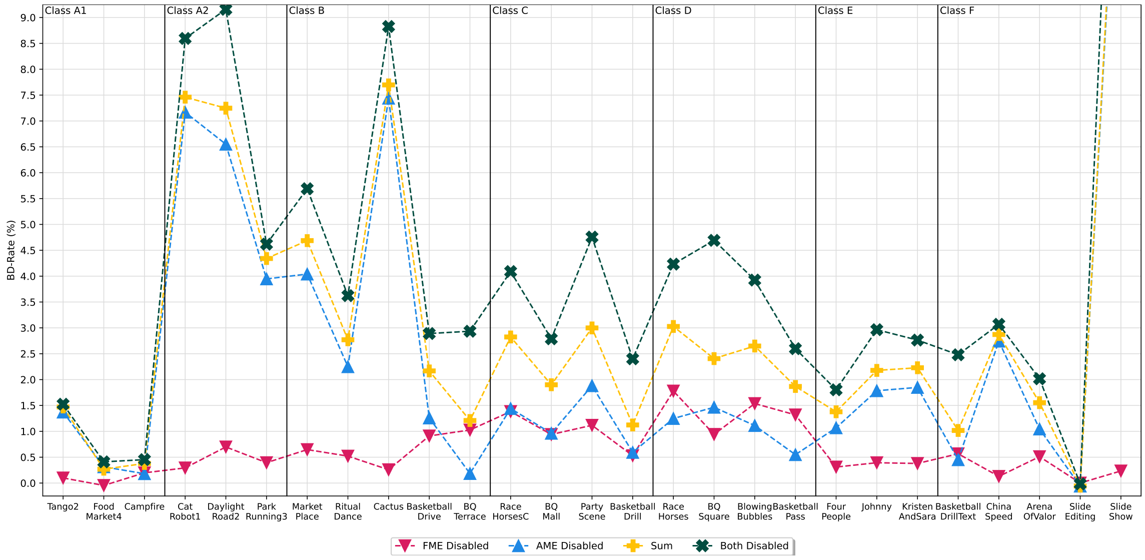
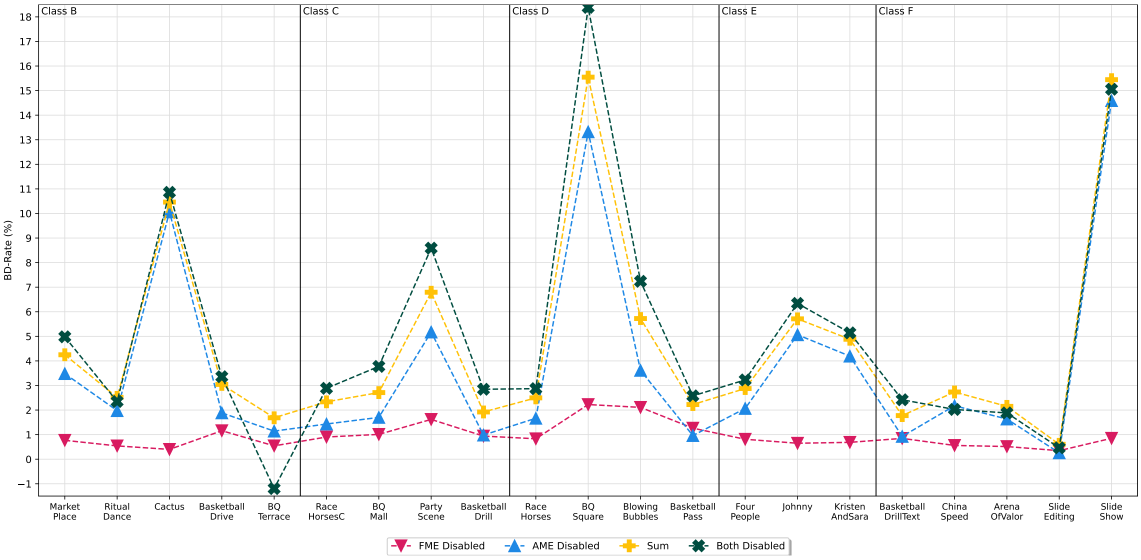


Table 4. LD temporal configuration BD-Rate for each sequence.

Sequence	FME Disabled	AME Disabled	Both Disabled	*Sum**
MarketPlace	0.768	3.483	4.980	4.250
RitualDance	0.541	1.976	2.348	2.517
Cactus	0.399	10.064	10.860	10.463
BasketballDrive	1.159	1.885	3.357	3.043
BQTerrace	0.543	1.143	-1.195	1.686
B Average	0.682	3.710	4.070	4.392
RaceHorsesC	0.903	1.432	2.885	2.335
BQMall	1.010	1.701	3.776	2.711
PartyScene	1.619	5.175	8.588	6.794
BasketballDrill	0.941	0.982	2.846	1.922
C Average	1.118	2.322	4.524	3.441
RaceHorses	0.831	1.666	2.871	2.496
BQSquare	2.224	13.323	18.385	15.547
BlowingBubbles	2.109	3.615	7.239	5.724
BasketballPass	1.263	0.962	2.578	2.224
D Average	1.607	4.891	7.768	6.498
FourPeople	0.811	2.065	3.226	2.876
Johnny	0.649	5.064	6.340	5.712
KristenAndSara	0.686	4.194	5.141	4.881
E Average	0.715	3.774	4.902	4.490
BasketballDrillText	0.847	0.927	2.419	1.775
ChinaSpeed	0.560	2.175	2.024	2.735
ArenaOfValor	0.515	1.635	1.883	2.150
SlideEditing	0.347	0.259	0.463	0.606
SlideShow	0.851	14.593	15.055	15.445
F Average	0.624	3.918	4.369	4.542
Average	0.932	3.729	5.051	4.662

****Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)

Figure 4. LD temporal configuration BD-Rate for each sequence.



Notice that the y-axis in Figure 3 represents a smaller BD-Rate interval than in Figure 4. Thus, in most cases, disabling a tool in RA has a lower impact than disabling it in LD. However, this largely depends on the sequence content. For instance, taking only the results of **Both Disabled** into consideration, the BD-Rate difference between RA and LD for the Cactus sequence is relatively small, 8.83% (RA) and 10.86% (LD), while there is a major difference for the BQSquare sequence, 4.69% (RA) and 18.38% (LD).

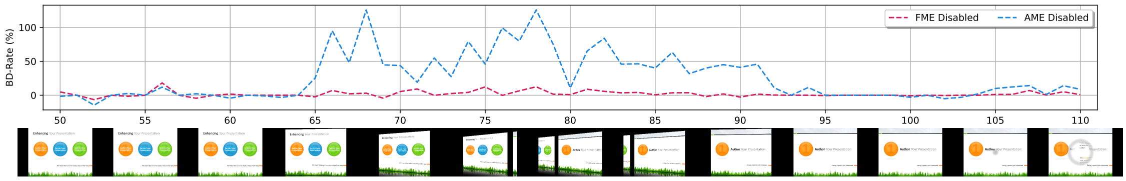
Nonetheless, for most sequences, in both RA and LD, AME has a larger impact than the FME, as can be noticed by the higher values of BD-Rate for **AME Disabled** than for **FME Disabled**. This result was expected, as being able to capture rotation, shearing, and zooming, the AME tool is versatile, bringing a significant enhancement in the coding efficiency of the majority of tested sequences.

As seen in both figures and tables, AME particularly enhances sequences:

- From class A2: CatRobot1 has an object that rotates, while DaylightRoad2 and ParkRunning3 have zooming;
- Cactus: similarly to CatRobot1, it has a rotating object;
- BQSquare: the camera is zooming and moving;
- SlideShow: the slide transitions present in this sequence are very characteristic affine transformations, such as shearing and zooming.

To show in greater detail the results of **FME Disabled** and **AME Disabled** for the SlideShow sequence in RA configuration, Figure 5 presents the BD-Rate computed for each frame of the sequence, along with a miniature of one in every 10 frames to provide a clue of the video content and how it relates to the BD-Rate. One may notice that when the sequence is around frame 65, the BD-Rate of **AME Disabled** starts to rise.

Figure 5. BD-Rate per frame for SlideShow sequence considering RA configuration, with the horizontal axis showing the frame index.



This corresponds to the beginning of a slide transition that ends around frame 95, when the BD-Rate of **AME Disabled** goes back to near zero. The transition in question is a prime example of an affine transformation. Thus, without the AME tool, the encoder expends a large amount of the available rate to signalize MVs, resulting in a sequence BD-Rate of 17.92%. Furthermore, the **FME Disabled** results in Figure 5 are almost negligible, as can also be noticed by the resulting sequence BD-Rate in Tables 3 (0.234%) and 4 (0.851%).

In fact, analyzing the overall **FME Disabled** results shows that the FME has negligible impacts when AME is enabled. Nevertheless, for low-resolution sequences, such as the ones from class D, the FME still has its importance. The effect can be more easily observed in Table 5, which presents the average increase in BD-Rate per class for each configuration tested.

Table 5. Average BD-Rate for each class.

Cfg.	Class	FME Disabled	AME Disabled	*Sum**	Both Disabled
Random Access (RA)	A1	0.084	0.622	0.706	0.796
	A2	0.462	5.888	6.349	7.457
	B	0.673	3.034	3.706	4.793
	C	0.992	1.220	2.212	3.509
	D	1.394	1.094	2.487	3.861
	E	0.361	1.567	1.928	2.511
	F	0.288	3.665	3.953	5.094
	Average	0.632	2.481	3.113	4.119
Low Delay (LD)	B	0.682	3.710	4.392	4.070
	C	1.118	2.322	3.441	4.524
	D	1.607	4.891	6.498	7.768
	E	0.715	3.774	4.490	4.902
	F	0.624	3.918	4.542	4.369
	Average	0.932	3.729	4.662	5.051

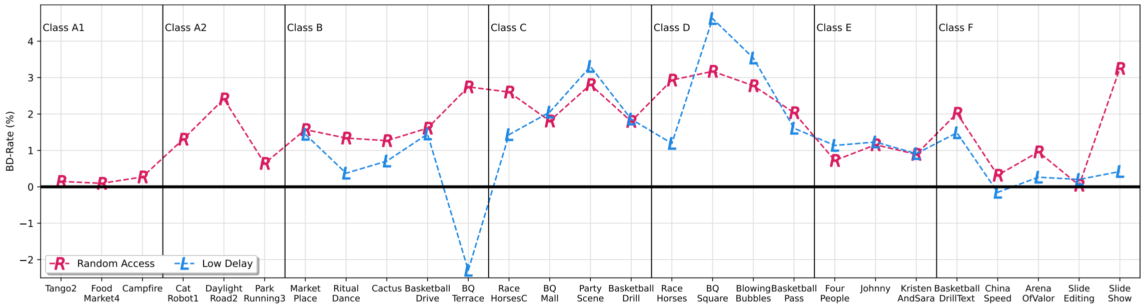
****Sum***=(BD-Rate FME Disabled) + (BD-Rate AME Disabled)

When only the FME is disabled, there is an average increase of 0.63% and 0.93% BD-Rate for RA and LD. The increase is bigger for classes C and D in both cases, corresponding to lower-resolution sequences. With a few exceptions in LD, the most striking being BQTerace, **Both Disabled** results in the highest BD-Rate values, which is expected. However, it is worth comparing the BD-Rate results from **Sum** with **Both Disabled**. Notice that in most cases, the **Sum** of BD-Rates is smaller than the BD-Rate of **Both Disabled**.

Considering the ME flow described in Figure 1 and this result, it is possible to conclude that AME is compensating for the lack of FME to some degree, which is in line with the hypothesis previously defined. On the other hand, even in light of this AME compensation, the impact of fully disabling the FME is still short of what was observed in previous standards and thus should be further studied.

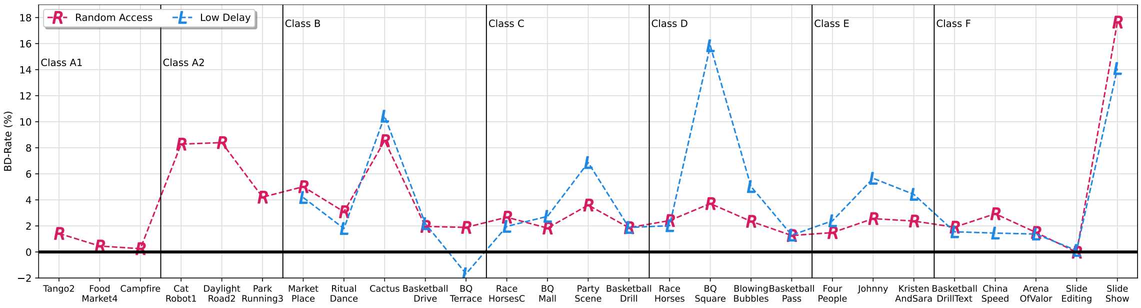
Another way of analyzing the FME effect is through Figure 6. The figure presents the BD-Rate for the sequences tested in both RA and LD configurations, but instead of using the usual reference of **Both Enabled**, **AME Disabled** is used as reference and compared to **Both Disabled**, *i.e.*, FME gains in a scenario where AME is not available.

Figure 6. Both Disabled BD-Rate using AME Disabled as reference.



In a similar fashion, Figure 7 shows the BD-Rate for AME using the same technique: using **FME Disabled** as the reference and comparing it to **Both Disabled**. Note the difference in y-axis scale for both cases, demonstrating a greater impact of AME in general.

Figure 7. Both Disabled BD-Rate using FME Disabled as reference.



Figures 3, 4, 6, and 7 present similar results, highlighting that even though the data is analyzed from a different point of view, the results found and analysis made so far are reasonable.

6. Conclusion

This work analyzed the coding efficiency of the Fractional Motion Estimation (FME) and Affine Motion Estimation (AME) within the VTM encoder, which is the reference implementation of the VVC standard. Firstly, our results show that disabling the FME in VTM causes a significantly smaller increase in BD-Rate relative to doing the same thing in the HM, the reference encoder of VVC predecessor, the HEVC, being reduced from 12% to less than 1%.

Also, the newly introduced AME tool enhances the coding efficiency of VTM by an average of 2.48% BD-Rate (RA) and 3.73% (LD). These results are compatible with the ones from the presented literature. Sequences that contain characteristic affine transformations have significant gains from AME, as shown by the entirety of class A2 and especially SlideShow.

By analyzing VVC ME flow and the results obtained from disabling both tools, it is also possible to conclude that AME can compensate for the loss of FME to some degree in the majority of tested sequences. Such a conclusion relies upon the fact that the BD-Rates from disabling both tools are larger than the sum of the BD-Rates of disabling them individually. Then again, the observed impact of disabling the FME still falls short of the expected.

This work sheds light on how tools interact and overlap during video coding, particularly in the ME step. Both FME and AME were analyzed, and data from multiple encodings were used to assess their impact on the coding efficiency of the VVC reference software. Furthermore, it was shown that analyzing tools working together provides valuable insights, revealing results that would not be possible by studying them in isolation.

Future work includes analyzing whether the 6-parameter AME is executed more times when skipping FME due to worse Rate-Distortion (RD) costs from IME alone and analyzing the effect of AMVR, so as to verify if the VTM is capable of noticing that no fractional MV is being found by TME.

References

Bossen, F., Boyce, J., Li, X., Seregin, V., and Sühring, K. (2020). Vtm common test conditions and software reference configurations for sdr video.

Bossen, F., Li, X., and Suehring, K. (2024). VVC Test Model (VTM) v23.1.

Bross, B., Wang, Y.-K., Ye, Y., Liu, S., Chen, J., Sullivan, G. J., and Ohm, J.-R. (2021). Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764.

Chien, W.-J., Zhang, L., Winken, M., Li, X., Liao, R.-L., Gao, H., Hsu, C.-W., Liu, H., and Chen, C.-C. (2021). Motion vector coding and block merging in the versatile video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3848–3861.

Filho, V. R. (2022). *Fixed search patterns and VLSI architecture for the efficient computation of the versatile video coding fractional motion estimation*. PhD thesis, Universidade Federal de Santa Catarina.

Filho, V. R., Monteiro, M., Seidel, I., Grellert, M., and Guntzel, J. L. (2021). Hardware-friendly search patterns for the versatile video coding fractional motion estimation.

Herglotz, C., Och, H., Meyer, A., Ramasubbu, G., Eichermüller, L., Kränzler, M., Brand, F., Fischer, K., Nguyen, D. T., Regensky, A., and Kaup, A. (2024). The Bjøntegaard bible why your way of comparing video codecs may be wrong. *IEEE Transactions on Image Processing*, 33:987–1001.

Herglotz, Christian (2024). bjontegaard v1.3.0.

ISO Central Secretary (2013). Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding. Standard ISO/IEC 23008-2, ISO, Geneva, CH.

ISO Central Secretary (2020). Information technology – Coded representation of immersive media – Part 3: Versatile video coding. Standard ISO/IEC 23090-3, ISO, Geneva, CH.

Liu, H., Zhang, L., Zhang, K., Xu, J., Wang, Y., Zhao, P., and Hong, D. (2019). JVET-M0255: AHG11: MMVD without fractional distances for SCC. , Bytedance Inc., Marrakech, MA.

Muñoz, M. M., Maass, D., Perleberg, M., Agostini, L., and Porto, M. (2023). Hardware design for the affine motion compensation of the VVC standard. page 1–4.

Sandvine (2024). Global internet phenomena report 2024.

Seidel, I. (2019). *Exploiting SATD properties to reduce energy in video coding*. PhD thesis, Universidade Federal de Santa Catarina.

Siqueira, I., Correa, G., and Grellert, M. (2021). Complexity and coding efficiency assessment of the Versatile Video Coding standard. 9217:1–5.

Sullivan, G., Ohm, J., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668.

Yang, H., Chen, H., Chen, J., Esenlik, S., Sethuraman, S., Xiu, X., Alshina, E., and Luo, J. (2021). Subblock-based motion derivation and inter prediction refinement in the versatile video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3862–3877.

Zhu, W., Liu, H., Zhang, K., Zhang, L., Xu, J., and Wang, Y. (2019). JVET-N0260-v1: Non-CE8: Adaptive fractional MVD search in DMVR for SCC. , Bytedance Inc., Geneva, CH.