

**UNIVERSIDADE FEDERAL DE PELOTAS**  
**Centro de Desenvolvimento Tecnológico**  
**Programa de Pós-Graduação em Computação**



Tese

**Provendo Eficiência Energética na Codificação de Vídeo usando Subtratores**  
**Aproximados na Predição Interquadros**

**Rafael dos Santos Ferreira**

Pelotas, 2024

**Rafael dos Santos Ferreira**

**Provendo Eficiência Energética na Codificação de Vídeo usando Subtratores  
Aproximados na Predição Interquadros**

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Bruno Zatt  
Coorientador: Prof. Dr. Cláudio Machado Diniz  
Colaborador: Prof. Dr. Luciano Volcan Agostini

Pelotas, 2024

Universidade Federal de Pelotas / Sistema de Bibliotecas  
Catalogação da Publicação

F384p Ferreira, Rafael dos Santos

Provendo eficiência energética na codificação de vídeo usando subtratores aproximados na predição interquadros [recurso eletrônico] / Rafael dos Santos Ferreira ; Bruno Zatt, orientador ; Cláudio Diniz, Luciano Agostini, coorientadores. — Pelotas, 2024.

173 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2024.

1. Subtrator. 2. TZS. 3. SAD. 4. Predição Inter. 5. SAD. I. Zatt, Bruno, orient. II. Diniz, Cláudio, coorient. III. Agostini, Luciano, coorient. IV. Título.

CDD 005

**Rafael dos Santos Ferreira**

**Provendo Eficiência Energética na Codificação de Vídeo usando Subtratores  
Aproximados na Predição Interquadros**

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

**Data da Defesa:** 03 de maio de 2024

**Banca Examinadora:**

Prof. Dr. Bruno Zatt (orientador)

Doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Cláudio Machado Diniz (coorientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Luciano Volcan Agostini (coorientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Nuno Roma

Doutor em Engenharia Electrotécnica e de Computadores pela Universidade Técnica de Lisboa.

Prof. Dr. Leonardo Bandeira Soares

Doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Marcelo Schiavon Porto

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Dedico esta tese a minha esposa, filhas e ao meu irmão Gabriel. . .

## AGRADECIMENTOS

Gostaria de expressar minha sincera apreciação a todos aqueles que estiveram ao meu lado durante essa jornada desafiadora, apoiando-me e compreendendo a minha ausência. Vocês foram verdadeiros pilares em minha vida, e sem o apoio de vocês, eu não teria alcançado esse marco significativo.

Agradeço em especial à minha amada esposa e filhas. Vocês suportaram minha ausência durante longas horas de estudo e dedicação à pesquisa. Sua compreensão e apoio incondicionais foram fundamentais para minha jornada acadêmica. Vocês foram meu refúgio e minha inspiração nos momentos mais difíceis.

Ao meu irmão Gabriel, sou imensamente grato por ter acreditado em mim desde o início. Seu incentivo constante e sua fé em minhas capacidades foram essenciais para que eu seguisse em frente, mesmo nos momentos de dúvida. Sua confiança foi um verdadeiro combustível para o meu sucesso.

Agradeço, também, aos meus pais, Mario e Fátima, que generosamente me apoiaram. Seus incentivos e ações fizeram uma grande diferença em minha jornada. Sua dedicação em tornar possível minha busca pelo conhecimento é verdadeiramente inspiradora.

Quero estender meus agradecimentos aos colegas de jornada, Alex Borges, Robson Domanski, Bruno Moura e Wagner Penny. Juntos compartilhamos desafios e triunfos, encorajando uns aos outros ao longo do caminho. Nossas conversas, trocas de ideias e colaborações foram inestimáveis e enriquecedoras. Vocês são uma parte valiosa da minha história acadêmica.

Gostaria de expressar minha gratidão a Cláudio Albano: amigo, professor e chefe. Ele foi a pessoa que me fez perceber que um problema de dicção não é uma limitação, o que me encorajou a também não me ver limitado. Seu apoio e suas atitudes foram extremamente motivadores. Um agradecimento especial é dedicado aos meus orientadores, Luciano Agostini, Cláudio Diniz e Bruno Zatt, pela compreensão e paciência demonstradas ao lidar com minha dupla jornada como empresário e pesquisador. Suas contribuições foram fundamentais em cada etapa deste caminho.

Luciano, seu encorajamento nas primeiras aulas foi decisivo. Agradeço pelo apoio quando apresentei o subtrator, um momento-chave que impulsionou minha trajetória. Sem sua motivação, talvez tudo isso não fosse possível.

Claudio, sua presença desde o mestrado foi vital. Agradeço por estar sempre disponível para me ouvir e orientar. Sua paciência e dedicação foram essenciais para o meu desenvolvimento acadêmico.

E a Bruno Zatt, meu profundo agradecimento por acreditar em mim e me aceitar como seu orientando. Sua orientação e apoio foram cruciais para a conclusão bem-sucedida deste doutorado.

Àqueles que não foram mencionados individualmente, meu sincero agradecimento também. Cada palavra de incentivo, cada gesto de apoio e cada pensamento positivo foram importantes para minha motivação e perseverança.



*Contra nós não resiste mal nenhum Pode mandar quem  
quiser Meu caminho é guiado pela fé.*

— BIGUP

## RESUMO

FERREIRA, Rafael dos Santos. **Provendo Eficiência Energética na Codificação de Vídeo usando Subtratores Aproximados na Predição Interquadros**. Orientador: Bruno Zatt. 2024. 173 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2024.

A eficiência energética na codificação de vídeo tem se tornado cada vez mais relevante, considerando a ampla utilização de vídeo em diversas aplicações modernas e o desafio associado à transmissão e armazenamento de grandes volumes de dados de vídeo. Nesse contexto, destaca-se o processo de codificação de vídeo, com especial atenção para a Predição Interquadros e a Soma das Diferenças Absolutas (*Sum of Absolute Differences* - SAD), fundamentais para a compressão eficiente de vídeo. A pesquisa apresentada nesta tese foca em aprimorar o cálculo de SAD através da implementação de subtratores aproximados, buscando um equilíbrio entre precisão e eficiência energética. Este trabalho se insere no panorama dos avanços tecnológicos em codificadores de vídeo, enfatizando a necessidade de soluções que diminuam o consumo energético sem impactar significativamente a qualidade do vídeo. As contribuições desta tese incluem o desenvolvimento de um subtrator aproximado inovador, batizado de *Approximate Subtractor* (AppS) e a implementação de um sistema de hardware dedicado à predição interquadros no padrão VVC (*Versatile Video Coding*).

Uma análise detalhada dos subtratores aproximados é realizada, incluindo o AppS. A pesquisa apresentada engloba uma avaliação abrangente desses subtratores, analisando seu impacto na qualidade de codificação e na eficiência energética, através de testes em diversos cenários e configurações de codificação.

Adicionalmente, o estudo explora a eficiência do algoritmo *Test Zone Search* (TZS) no contexto dos subtratores aproximados, examinando como influenciam na precisão dos vetores de movimento e no processo de codificação de forma geral.

No estudo sobre a eficiência energética e o desempenho das operações de Soma das Diferenças Absolutas (SAD) em subtratores aproximados, os resultados variaram significativamente entre os modelos analisados. O subtrator AppS1 demonstrou consistência e alta eficácia, com desempenho constante e redução significativa no consumo energético. Em contraste, o AXCS24 registrou um aumento nas operações de SAD e maior consumo de energia, destacando a importância de uma seleção cuidadosa dos subtratores para otimizar tanto a eficiência energética quanto a precisão na codificação de vídeo.

Palavras-chave: Subtrator aproximado SAD TZS Predição Inter

## ABSTRACT

FERREIRA, Rafael dos Santos. **Providing Energy Efficiency in Video Coding Using Approximate Subtractors in Interframe Prediction.** Advisor: Bruno Zatt. 2024. 173 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2024.

Energy efficiency in video encoding has become increasingly relevant given the widespread use of video in various modern applications and the associated challenge of transmitting and storing large volumes of video data. In this context, the video encoding process is highlighted, with special attention to Interframe Prediction and the Sum of Absolute Differences (SAD), which are fundamental for efficient video compression. The research presented in this thesis focuses on enhancing the SAD calculation through the implementation of approximate subtractors, seeking a balance between precision and energy efficiency. This work is part of the panorama of technological advancements in video encoders, emphasizing the need for solutions that reduce energy consumption without significantly impacting video quality. The contributions of this thesis include the development of an innovative approximate subtractor, named Approximate Subtractor (AppS), and the implementation of a hardware system dedicated to interframe prediction in the Versatile Video Coding (VVC) standard.

A detailed analysis of approximate subtractors is carried out, including the AppS. The presented research encompasses a comprehensive evaluation of these subtractors, analyzing their impact on encoding quality and energy efficiency, through tests in various scenarios and encoding settings.

Additionally, the study explores the efficiency of the Test Zone Search (TZS) algorithm in the context of approximate subtractors, examining how they influence the accuracy of motion vectors and the overall encoding process.

In the study on the energy efficiency and performance of the Sum of Absolute Differences (SAD) operations in approximate subtractors, the results varied significantly among the analyzed models. The AppS1 subtractor demonstrated consistency and high efficacy, with constant performance and significant reduction in energy consumption. In contrast, the AXCS24 recorded an increase in SAD operations and greater energy consumption, highlighting the importance of careful selection of subtractors to optimize both energy efficiency and precision in video encoding, as detailed in the tables of the presented thesis.

Keywords: Subtractor Imprecise SAD TZS Inter Prediction

## LISTA DE FIGURAS

|           |                                                                                                             |    |
|-----------|-------------------------------------------------------------------------------------------------------------|----|
| Figura 1  | Codificador de Vídeo Genérico (Tanou; Blestel, 2019) . . . . .                                              | 29 |
| Figura 2  | Divisão de Blocos no HEVC e VVC (Sjöberg et al., 2020) . . . . .                                            | 33 |
| Figura 3  | Esquema de Particionamento de Blocos no VVC (Gao et al., 2020) . . . . .                                    | 34 |
| Figura 4  | Exemplificação da Intra-predição em um Bloco. (Pfaff et al., 2020) . . . . .                                | 35 |
| Figura 5  | Modos de Predição Intra. (Pfaff et al., 2020) . . . . .                                                     | 36 |
| Figura 6  | Blocos de Predição. (Correa, 2014) . . . . .                                                                | 36 |
| Figura 7  | Algoritmo de Busca Completa. (Marcelo Porto, 2012) . . . . .                                                | 42 |
| Figura 8  | Algoritmo de Busca Diamond. (Marcelo Porto, 2012) . . . . .                                                 | 43 |
| Figura 9  | Algoritmo de busca Three Step Search. (Marcelo Porto, 2012) . . . . .                                       | 44 |
| Figura 10 | Etapas do Test Zone Search. Adaptado de (Doan et al., 2017) . . . . .                                       | 48 |
| Figura 11 | Preditores do TZS . . . . .                                                                                 | 49 |
| Figura 12 | Busca Inicial do TZS. (Doan et al., 2017) . . . . .                                                         | 50 |
| Figura 13 | Busca Dois Pontos . (Doan et al., 2017) . . . . .                                                           | 50 |
| Figura 14 | Busca Varredura. (Doan et al., 2017) . . . . .                                                              | 51 |
| Figura 15 | Somador Aproximado GeAr (Shafique et al., 2015). Adaptado: (Roger Endrigo Carvalho Porto, 2020) . . . . .   | 59 |
| Figura 16 | Somador Aproximado Completo 2PSA (Porto et al., 2020) . . . . .                                             | 59 |
| Figura 17 | Somador Aproximado 2PSA (Porto et al., 2020) . . . . .                                                      | 60 |
| Figura 18 | Somador Aproximado LOA (Mahdiani et al., 2010). . . . .                                                     | 60 |
| Figura 19 | Somador Aproximado ETA-I (Mahdiani et al., 2010). Adaptado: (Roger Endrigo Carvalho Porto, 2020) . . . . .  | 61 |
| Figura 20 | Somador Aproximado ETA II (Mahdiani et al., 2010). Adaptado: (Roger Endrigo Carvalho Porto, 2020) . . . . . | 62 |
| Figura 21 | Somador Aproximado Copy. Adaptado por (Macedo et al., 2018). . . . .                                        | 62 |
| Figura 22 | Somador Aproximado ACAA (Kahng; Kang, 2012). Adaptado: (Roger Endrigo Carvalho Porto, 2020) . . . . .       | 63 |
| Figura 23 | Somador Aproximado 1-bit FA (Kedem et al., 2011). . . . .                                                   | 63 |
| Figura 24 | Subtrator AXSC 1 (Chen et al., 2015). . . . .                                                               | 64 |
| Figura 25 | Subtrator AXSC 2 (Chen et al., 2015). . . . .                                                               | 65 |
| Figura 26 | Subtrator AXSC 3 (Chen et al., 2015). . . . .                                                               | 65 |
| Figura 27 | Subtrator ApXs (Muthulakshmi; Dash; Prabakaran, 2018). . . . .                                              | 65 |
| Figura 28 | Subtrator APXS 2 (Muthulakshmi; Dash; Prabakaran, 2018). . . . .                                            | 66 |
| Figura 29 | Subtrator AXS 1 (Reddy et al., 2018). . . . .                                                               | 66 |
| Figura 30 | Subtrator AXS 2 (Reddy et al., 2018). . . . .                                                               | 66 |
| Figura 31 | Subtrator AXS 3 (Reddy et al., 2018). . . . .                                                               | 67 |
| Figura 32 | Subtrator AppS de 1-bit. . . . .                                                                            | 69 |

|           |                                                                                                               |     |
|-----------|---------------------------------------------------------------------------------------------------------------|-----|
| Figura 33 | Exemplo de Configuração do AppS com 4-bits. . . . .                                                           | 69  |
| Figura 34 | Análise Inicial . . . . .                                                                                     | 75  |
| Figura 35 | Classificação dos Subtratores por Resultado do Cálculo de SAD.<br>Todos os Blocos . . . . .                   | 84  |
| Figura 36 | Classificação dos Vetores de Movimento por Subtrator Aproximado.<br>Todos os Tamanhos de Bloco . . . . .      | 85  |
| Figura 37 | Classificação dos Subtratores por Resultado do Cálculo de SAD.<br>Blocos 32x32 . . . . .                      | 106 |
| Figura 38 | Classificação dos Vetores de Movimento por Subtrator Aproximado.<br>Blocos 32x32 . . . . .                    | 108 |
| Figura 39 | Classificação dos Subtratores pela Média das Diferenças Absolutas<br>dos Resultado do Cálculo de SAD. . . . . | 111 |
| Figura 40 | Classificação da Média de todos Vetores de Movimento por Subtra-<br>tor Aproximado . . . . .                  | 113 |
| Figura 41 | Arquitetura do TZS . . . . .                                                                                  | 117 |
| Figura 42 | Metodologia de Síntese . . . . .                                                                              | 118 |
| Figura 43 | Arquitetura do Posições . . . . .                                                                             | 119 |
| Figura 44 | Módulo de Corte . . . . .                                                                                     | 121 |
| Figura 45 | Arquitetura do SAD para blocos 32x32 . . . . .                                                                | 122 |
| Figura 46 | <i>Máquina de Estados TZS</i> . . . . .                                                                       | 125 |
| Figura 47 | Classificação dos Subtratores por Resultado do Cálculo de SAD.<br>Blocos 16x16 . . . . .                      | 151 |
| Figura 48 | Classificação dos Vetores de Movimento por Subtrator Aproximado.<br>Blocos 16x16 . . . . .                    | 152 |
| Figura 49 | Classificação dos Subtratores por Resultado do Cálculo de SAD.<br>Blocos 64x64 . . . . .                      | 158 |
| Figura 50 | Classificação dos Vetores de Movimento por Subtrator Aproximado.<br>Blocos 64x64 . . . . .                    | 159 |
| Figura 51 | Classificação dos Subtratores por Resultado do Cálculo de SAD.<br>Blocos 128x128 . . . . .                    | 165 |
| Figura 52 | Classificação dos Vetores de Movimento por Subtrator Aproximado.<br>Blocos 64x64 . . . . .                    | 167 |
| Figura 53 | Distribuição Entre Etapas TZS Blocos 32x32 . . . . .                                                          | 173 |

## LISTA DE TABELAS

|           |                                                                                                                                                                                                                                                                      |    |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1  | Tabela Verdade de um Subtrator Preciso e do AppS . . . . .                                                                                                                                                                                                           | 70 |
| Tabela 2  | Tabela Verdade dos Subtratores Aproximados . . . . .                                                                                                                                                                                                                 | 71 |
| Tabela 3  | Precisão de Subtrações . . . . .                                                                                                                                                                                                                                     | 71 |
| Tabela 4  | Resultados da Síntese dos Subtratores . . . . .                                                                                                                                                                                                                      | 72 |
| Tabela 5  | Sumarização dos Resultados . . . . .                                                                                                                                                                                                                                 | 73 |
| Tabela 6  | SI e TI dos Vídeos Analisados, Considerando os 16 Primeiros Frames                                                                                                                                                                                                   | 75 |
| Tabela 7  | Contagem de Blocos Usados na Codificação dos Vídeos <i>BQTerrace</i> e <i>Cactus</i> Usando o Subtrator <i>Baseline</i> . . . . .                                                                                                                                    | 79 |
| Tabela 8  | Contagem de Blocos Utilizando Subtratores Aproximados . . . . .                                                                                                                                                                                                      | 88 |
| Tabela 9  | Representatividade na Escolha de Blocos. . . . .                                                                                                                                                                                                                     | 89 |
| Tabela 10 | Contagem de SAD com Subtrator <i>Baseline</i> para todos os Tamanhos de Bloco em cada uma das Etapas do TZS . . . . .                                                                                                                                                | 89 |
| Tabela 11 | Quantidade Percentual de SADs Executados em Comparação com o Subtrator <i>Baseline</i> . Valores Negativos Sinalizam uma Redução nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1,570,066. . . . .                                                                   | 89 |
| Tabela 12 | Comparação da Quantidade de SADs Executadas: Subtrator <i>Baseline</i> vs. Subtrator Aproximado no Vídeo Cactus, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo: Cactus, TI: 6,056 e SI: 1.570,066 . . . . . | 89 |
| Tabela 13 | Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator <i>Baseline</i> e Subtratores Aproximados para Todos os Tamanhos de Blocos . . . . .                                                                                                           | 90 |
| Tabela 14 | Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator <i>Baseline</i> e Subtratores Aproximados para Todos os Tamanhos de Blocos . . . . .                                                                                                    | 90 |
| Tabela 15 | Resultados de BD-RATE. Blocos de Todos os Tamanhos . . . . .                                                                                                                                                                                                         | 90 |
| Tabela 16 | Contagem de SAD com Subtrator <i>Baseline</i> para Tamanhos de Bloco 32x32 em Cada uma das Etapas do TZS . . . . .                                                                                                                                                   | 91 |
| Tabela 17 | Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução 480p . . . . .                                   | 94 |
| Tabela 18 | Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução:720 . . . . .                                    | 97 |

|           |                                                                                                                                                                                                                                                          |     |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela 19 | Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo KristenAndSara, TI:2,148 , SI: 7247,037. | 98  |
| Tabela 20 | Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução 1080p. . . . .                       | 100 |
| Tabela 21 | Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. . . . .                                        | 101 |
| Tabela 22 | Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução 4K . . . . .                         | 104 |
| Tabela 23 | Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução 4k . . . . .                         | 105 |
| Tabela 24 | Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32 . . . . .                                                                                                                        | 106 |
| Tabela 25 | Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32 .                                                                                                                         | 107 |
| Tabela 26 | Eficiência BD-RATE. Blocos 32x32 . . . . .                                                                                                                                                                                                               | 109 |
| Tabela 27 | Sumário da Contagem de SAD para Subtratores Aproximados em Diferentes Tamanhos de Blocos . . . . .                                                                                                                                                       | 110 |
| Tabela 28 | Sumarização da Diferença Absoluta do Resultado de SAD entre o Subtrator Aproximado e o Subtrator Baseline . . . . .                                                                                                                                      | 111 |
| Tabela 29 | Sumarização da Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32 . . . . .                                                                                                  | 112 |
| Tabela 30 | Eficiência Média Total de Codificação em BD-Rate . . . . .                                                                                                                                                                                               | 114 |
| Tabela 31 | Comparação entre o Total de SADS Executados e a Eficiência de Codificação em BD-Rate . . . . .                                                                                                                                                           | 114 |
| Tabela 32 | Interface Modulo Posição . . . . .                                                                                                                                                                                                                       | 119 |
| Tabela 33 | Interfaces Módulo de Corte . . . . .                                                                                                                                                                                                                     | 121 |
| Tabela 34 | Interface SAD . . . . .                                                                                                                                                                                                                                  | 122 |
| Tabela 35 | Resultado Síntese do Módulo de SAD para frequência de 166 GHz e Potência em ( $\mu$ W) . . . . .                                                                                                                                                         | 124 |
| Tabela 36 | Interfaces Modulo TZS . . . . .                                                                                                                                                                                                                          | 126 |
| Tabela 37 | Resultados de Potência ( $\mu$ W) Frequência para frequência de 142 Mhz                                                                                                                                                                                  | 127 |
| Tabela 38 | Desempenho dos Subtratores na Contagem de SADs Executados e Resultado de Potencia ( $\mu$ W) e Energia(nJ) . . . . .                                                                                                                                     | 129 |
| Tabela 39 | Contagem de SAD com Subtrator Exato para tamanhos de bloco 16x16 em cada uma das etapas do TZS . . . . .                                                                                                                                                 | 148 |

|           |                                                                                                                                                                                                                                                        |     |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela 40 | Comparação da Quantidade de SADs Executadas para blocos 16x16: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066.   | 149 |
| Tabela 41 | Comparação da Quantidade de SADs Executadas para blocos 16x16: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066    | 150 |
| Tabela 42 | Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 16x16 . . . . .                                                                                                                      | 150 |
| Tabela 43 | Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 16x16 .                                                                                                                       | 152 |
| Tabela 44 | Eficiência BD-RATE. Blocos 16x16 . . . . .                                                                                                                                                                                                             | 153 |
| Tabela 45 | Contagem de SAD com Subtrator Exato para blocos 64x64 em cada uma das etapas do TZS . . . . .                                                                                                                                                          | 156 |
| Tabela 46 | Comparação da Quantidade de SADs Executadas para blocos 64x64: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066    | 156 |
| Tabela 47 | Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066 .  | 157 |
| Tabela 48 | Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 64x64 . . . . .                                                                                                                      | 157 |
| Tabela 49 | Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 64x64 .                                                                                                                       | 159 |
| Tabela 50 | Eficiência BD-RATE. Blocos 64x64 . . . . .                                                                                                                                                                                                             | 161 |
| Tabela 51 | Contagem de SAD com Subtrator Exato para Blocos 128x128 em Cada uma das Etapas do TZS . . . . .                                                                                                                                                        | 162 |
| Tabela 52 | Comparação da Quantidade de SADs Executadas para blocos 128x128: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066. | 164 |
| Tabela 53 | Comparação da Quantidade de SADs Executadas para blocos 128x128: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066. | 164 |
| Tabela 54 | Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 128x128 . . . .                                                                                                                      | 165 |
| Tabela 55 | Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 128x128                                                                                                                       | 167 |
| Tabela 56 | Eficiência BD-RATE. Blocos 128x128 . . . . .                                                                                                                                                                                                           | 168 |
| Tabela 57 | Variação da Imprecisão no Subtrator AppS variando os Bits Imprecisos                                                                                                                                                                                   | 169 |
| Tabela 58 | Variação da Imprecisão no Subtrator AppS variando os Bits Imprecisos                                                                                                                                                                                   | 170 |
| Tabela 59 | Variação da Imprecisão no Subtrator AppS variando os Bits Imprecisos                                                                                                                                                                                   | 171 |



## LISTA DE ABREVIATURAS E SIGLAS

|        |                                                  |
|--------|--------------------------------------------------|
| ACA    | <i>Almost Correct Adder</i>                      |
| ACAA   | <i>Accuracy Configurable Approximate Adder</i>   |
| AFF    | <i>AFFine motion compensation</i>                |
| ALF    | <i>Adaptive Loop Filter</i>                      |
| AME    | <i>Affine Motion Estimation</i>                  |
| AMVR   | <i>Adaptive MV Resolution</i>                    |
| AppS   | <i>Approximate Subtractor</i>                    |
| AR     | <i>Realidade Aumentada</i>                       |
| AXCS   | <i>An Exact Subtractor Cell</i>                  |
| AXS    | <i>ApproXimate Subtractor</i>                    |
| BDOF   | <i>BiDirectional Optical Flow</i>                |
| BT     | <i>Binary Tree</i>                               |
| CABAC  | <i>Context-Adaptive Binary Arithmetic Coding</i> |
| CC-ALF | <i>Cross-Component Adaptive Loop Filter</i>      |
| CCLM   | <i>Cross Component Linear Model</i>              |
| CIIP   | <i>Combined Intra-Inter Prediction</i>           |
| CLA    | <i>Carry Look-ahead Adder</i>                    |
| CM     | <i>Compensação de Movimento</i>                  |
| CMOS   | <i>Complementary Metal-Oxide-Semiconductor</i>   |
| CTU    | <i>Coding Tree Unit</i>                          |
| CTUs   | <i>Coding Tree Units</i>                         |
| CU     | <i>Coding Unit</i>                               |
| CUs    | <i>Coding Units</i>                              |
| DA     | <i>Distância Absoluta</i>                        |
| DBF    | <i>Deblocking Filter</i>                         |
| DCT    | <i>Discrete Cosine Transform</i>                 |

|         |                                                                                                 |
|---------|-------------------------------------------------------------------------------------------------|
| DMVR    | <i>Decoder-side Motion Vector Refinement</i>                                                    |
| DQ      | <i>Dependent Quantization</i>                                                                   |
| DS      | <i>Diamond Search</i>                                                                           |
| DSP     | <i>Digital Signal Processing</i>                                                                |
| DST     | <i>Discrete Sine Transform</i>                                                                  |
| DST-VII | <i>Type VII Discrete Sine Transform</i>                                                         |
| EM      | <i>Estimação de Movimento</i>                                                                   |
| ETA     | <i>Error Tolerant Adder I</i>                                                                   |
| EXSC    | <i>EXact Subtractor Cell</i>                                                                    |
| FA      | <i>Full Adder</i>                                                                               |
| FME     | <i>Fractional Motion Estimation</i>                                                             |
| FS      | <i>Full Search</i>                                                                              |
| GBI     | <i>Generalized BiPrediction</i>                                                                 |
| GDR     | <i>Gradual Decoding Refresh</i>                                                                 |
| GeAr    | <i>Generic Accuracy Configurable Adder</i>                                                      |
| GOP     | <i>Group of Pictures</i>                                                                        |
| GPM     | <i>Geometric Partitioning Mode</i>                                                              |
| HDR     | <i>High Dynamic Range</i>                                                                       |
| HEVC    | <i>High Efficiency Video Coding</i>                                                             |
| HMVP    | <i>History-Based MV Prediction</i>                                                              |
| IA      | <i>Inteligência Artificial</i>                                                                  |
| IBGE    | <i>Instituto Brasileiro de Geografia e Estatística</i>                                          |
| ISO/IEC | <i>International Organization for Standardization/International Electrotechnical Commission</i> |
| ISP     | <i>Intra Sub-Partitions</i>                                                                     |
| ITU-T   | <i>International Telecommunication Union</i>                                                    |
| LDSP    | <i>Large Diamond Search Pattern</i>                                                             |
| LFNST   | <i>Low-Frequency Non-separable Secondary Transform</i>                                          |
| LMCS    | <i>Luma Mapping with Chroma Scaling</i>                                                         |
| LOA     | <i>Lower-art-OR</i>                                                                             |
| LSB     | <i>Least Significant Bit</i>                                                                    |
| LUT     | <i>Lookup Table</i>                                                                             |
| MED     | <i>Mean Error Distance</i>                                                                      |
| MIP     | <i>Matrix-based Intra Prediction</i>                                                            |

|                   |                                                    |
|-------------------|----------------------------------------------------|
| ML                | <i>Machine Learning</i>                            |
| MMVD              | <i>Merge Mode on Motion Vector Difference</i>      |
| MPEG              | <i>Motion Picture Experts Group</i>                |
| MRED              | <i>Mean Relative Error Distance</i>                |
| MRL               | <i>Multiple Reference Line Intra Prediction</i>    |
| MSB               | <i>Most Significant Bit</i>                        |
| MSE               | <i>Mean Squared Error</i>                          |
| MtCO <sub>2</sub> | <i>Metric Tonnes of Carbon Dioxide</i>             |
| MTS               | <i>Multiple Transform Selection</i>                |
| MTT               | <i>Multi-Type Tree</i>                             |
| MVD               | <i>Motion Vector Difference</i>                    |
| MVP               | <i>Motion Vector Prediction</i>                    |
| NED               | <i>Normalized Error Distance</i>                   |
| PA                | <i>Probabilidade Aceita</i>                        |
| PDPC              | <i>Position-Dependent Prediction Combination</i>   |
| PMA               | <i>Precisão Mínima Aceitável</i>                   |
| PMTZS             | <i>Predição de Vetor de Movimento no TZS</i>       |
| Pre               | <i>Precisão</i>                                    |
| PSNR              | <i>Peak Signal-to-Noise Ratio</i>                  |
| PU <sub>s</sub>   | <i>Prediction Units</i>                            |
| QP                | <i>Quantization Parameter</i>                      |
| QT                | <i>Quadtree</i>                                    |
| QTMT              | <i>Quadtree with nested Multi-Type Tree</i>        |
| RBS               | <i>Ripple Borrow Subtractor</i>                    |
| RCA               | <i>Ripple Carry Adder</i>                          |
| RDO               | <i>Rate-Distortion Optimization</i>                |
| Ref               | <i>Busca Refinada</i>                              |
| SAD               | <i>Sum of Absolute Differences</i>                 |
| SADT              | <i>Sum of Transformed Absolute Differences</i>     |
| SAO               | <i>Sample Adaptive Offset</i>                      |
| SATD              | <i>Soma das Diferenças Transformadas Absolutas</i> |
| SBT               | <i>Sub-Block Transform</i>                         |
| SbTMVSubPu        | <i>Temporal Motion Vector Prediction</i>           |
| SDSP              | <i>Small Diamond Search Pattern</i>                |

|       |                                           |
|-------|-------------------------------------------|
| SMVD  | <i>Symmetric Motion Vector Difference</i> |
| SRFFT | <i>Split Radix Fast Fourier Transform</i> |
| SSE   | <i>Soma das Diferenças Quadráticas</i>    |
| SW    | <i>Search Window</i>                      |
| TSS   | <i>Three Step Search</i>                  |
| TT    | <i>Ternary Tree</i>                       |
| TUs   | <i>Transform Units</i>                    |
| TZS   | <i>Test Zone Search</i>                   |
| VCEG  | <i>Variable Latency Speculative Adder</i> |
| VLSI  | <i>Very-Large-Scale Integration</i>       |
| VM    | <i>Vector of Movement</i>                 |
| VR    | <i>Virtual Reality</i>                    |
| VTM   | <i>Versatile Test Model</i>               |
| VVC   | <i>Versatile Video Coding</i>             |
| XOR   | <i>Exclusive OR</i>                       |

# SUMÁRIO

|          |                                                             |    |
|----------|-------------------------------------------------------------|----|
| <b>1</b> | <b>INTRODUÇÃO</b>                                           | 23 |
| 1.1      | Tese                                                        | 26 |
| 1.2      | Questões de Pesquisa                                        | 26 |
| 1.3      | Organização do Texto                                        | 26 |
| <b>2</b> | <b>A CODIFICAÇÃO DE VÍDEO</b>                               | 28 |
| 2.1      | O Codificador de Vídeo Genérico                             | 29 |
| 2.2      | Padrão <i>Versatile Video Coding</i> - VVC                  | 31 |
| 2.2.1    | Estrutura de Particionamento                                | 33 |
| 2.2.2    | Predição Intra-Quadro                                       | 34 |
| 2.2.3    | Predição Inter-Quadros                                      | 36 |
| 2.2.4    | Transformadas                                               | 37 |
| 2.2.5    | Quantização                                                 | 38 |
| 2.2.6    | Codificação de Entropia                                     | 39 |
| 2.2.7    | Filtros de Laço                                             | 39 |
| 2.3      | Estimação de Movimento                                      | 41 |
| 2.3.1    | Métricas de Distorção                                       | 44 |
| 2.4      | Algoritmo Test Zone Search                                  | 46 |
| 2.4.1    | Busca Inicial                                               | 49 |
| 2.4.2    | Busca de Dois Pontos                                        | 50 |
| 2.4.3    | A Etapa de Busca de Varredura ( <i>Raster</i> )             | 51 |
| 2.4.4    | Busca Refinada                                              | 51 |
| <b>3</b> | <b>COMPUTAÇÃO APROXIMADA</b>                                | 53 |
| 3.1      | Níveis de Abstração                                         | 55 |
| 3.2      | Operadores Aritméticos Aproximados                          | 57 |
| 3.2.1    | Somadores Aproximados                                       | 57 |
| 3.2.2    | Subtratores Aproximados                                     | 63 |
| <b>4</b> | <b>APPROXIMATE SUBTRACTOR: UM NOVO SUBTRATOR</b>            | 68 |
| 4.0.1    | Comparando AppS com Outros Subtratores                      | 70 |
| <b>5</b> | <b>ANÁLISE DO ALGORITMO TZS COM SUBTRATORES APROXIMADOS</b> | 74 |
| 5.1      | Escolha do Tamanho de Blocos                                | 78 |
| 5.1.1    | Representatividade                                          | 80 |
| 5.2      | Análise de Sensibilidade à Aproximação                      | 81 |
| 5.2.1    | Contagem de SAD executados: Todos os Blocos                 | 81 |
| 5.2.2    | Análise de Blocos 32x32                                     | 86 |

|            |                                                                                  |     |
|------------|----------------------------------------------------------------------------------|-----|
| <b>5.3</b> | <b>Sumarização de todos os Resultados:Desempenho Geral dos Sub-<br/>tratores</b> | 109 |
| 5.3.1      | Totais de SADs Executados                                                        | 109 |
| 5.3.2      | Média Geral dos Resultados de SAD                                                | 110 |
| 5.3.3      | Distância dos Vetores de Movimento                                               | 112 |
| 5.3.4      | Eficiência de Codificação                                                        | 113 |
| <b>6</b>   | <b>ARQUITETURA DO TZS</b>                                                        | 116 |
| 6.1        | Metodologia de Síntese                                                           | 117 |
| 6.2        | Módulo Posições                                                                  | 118 |
| 6.3        | Módulo Corte                                                                     | 120 |
| 6.4        | Módulo SAD                                                                       | 121 |
| 6.5        | Módulo de Controle                                                               | 124 |
| 6.6        | Arquitetura TZS: Módulo Topo                                                     | 126 |
| 6.7        | Discussão dos Resultados                                                         | 127 |
| 6.8        | Discussão Frente aos Trabalhos Relacionados                                      | 129 |
| <b>7</b>   | <b>CONCLUSÃO</b>                                                                 | 131 |
| 7.1        | Trabalhos Futuros                                                                | 133 |
|            | <b>REFERÊNCIAS</b>                                                               | 134 |
|            | <b>APÊNDICE A PUBLICAÇÕES</b>                                                    | 146 |
|            | <b>APÊNDICE B ANÁLISE PARA BLOCOS DE TAMANHOS FIXO</b>                           | 147 |
| <b>B.1</b> | <b>Contagem de SAD executados:Blocos 16x16</b>                                   | 147 |
| B.1.1      | Análise do Resultado do Resultado do Cálculo do SAD para blocos 16x16            | 150 |
| B.1.2      | Análise da Distância do Bloco 16x16                                              | 151 |
| B.1.3      | Eficiência de Codificação para blocos 16x16                                      | 153 |
| <b>B.2</b> | <b>Contagem de SAD executados:Blocos 64x64</b>                                   | 154 |
| B.2.1      | Análise do Resultado do Resultado do Cálculo do SAD para blocos 64x64            | 156 |
| B.2.2      | Análise Distância para Blocos 64x64                                              | 158 |
| B.2.3      | Eficiência de Codificação para blocos 64x64                                      | 160 |
| B.2.4      | Contagem de SAD executados:Blocos 128x128                                        | 160 |
| B.2.5      | Análise do Resultado do Resultado do Cálculo do SAD para blocos<br>128x128       | 163 |
| B.2.6      | Eficiência de Codificação para blocos 128x128                                    | 167 |
| <b>B.3</b> | <b>Máscara de Imprecisão</b>                                                     | 169 |
| <b>B.4</b> | <b>Distribuição Entre Etapas TZS</b>                                             | 172 |

# 1 INTRODUÇÃO

Em 2019, cerca de 4,4 bilhões de pessoas acessaram a internet através de dispositivos móveis, o que representa 56% da população mundial (Clement, 2019). O mercado multimídia, impulsionado por empresas como *Google* e *Netflix*, tem mostrado um crescimento continuado, com a *Netflix* ultrapassando 200 milhões de assinantes ao redor do mundo (Stoll, 2021). No Brasil, *smartphones* e televisões são os principais dispositivos utilizados para acesso à internet. O uso da televisão no Brasil para acesso à internet teve um aumento de 4,2 vezes desde 2021 (Ibge, 2021). Além disso, a pandemia da COVID-19 acelerou o consumo de vídeo, com um aumento de 118% nas visualizações e 380% nos *downloads* em março de 2020, comparado a 2019 (Lederer, 2020).

A crescente utilização de serviços multimídia tem intensificado o tráfego de dados pela internet. Atualmente, os vídeos digitais representam aproximadamente 80% do tráfego total, um reflexo da sua alta demanda por dados. Para ilustrar, um vídeo em HD com duração de 10 horas contém mais informações do que todos os artigos da *Wikipedia* em inglês (Project, 2020). Em 2023, observou-se que 70% das visualizações no YouTube originaram-se de dispositivos móveis, e estima-se que até 2025 o tráfego móvel cresça mais 25%, impulsionado principalmente pelo consumo crescente de vídeos (Rating, 2024). Além disso, em 2018, o tráfego de vídeos digitais foi responsável pela geração de aproximadamente 300 milhões de toneladas de CO<sub>2</sub>, uma quantidade comparável às emissões anuais de carbono da Espanha (Project, 2020).

Nesse contexto, a codificação de vídeo é uma tecnologia crucial para o funcionamento do ecossistema de distribuição de vídeos digitais. Trata-se de um processo responsável por reduzir o volume de dados necessários para representar vídeos facilitando o armazenamento e a transmissão. Desse modo, é necessário um processo de decodificação do vídeo para sua exibição ao usuário. O objetivo dos codificadores é prover alta eficiência de compressão codificação, ou seja, reduzir o volume de dados mantendo a melhor qualidade de imagem possível. O aumento da eficiência de codificação provida por cada nova geração de codificadores, no entanto, vem ao custo de aumento no esforço computacional e no consumo de energia associado.

O elevado esforço computacional é particularmente desafiador para dispositivos embarcados que enfrentam limitações em termos de energia e dissipação de potência (Hennessy; Patterson, 2017). Para atender aos requisitos de processamento de codificadores de vídeo, especialmente para sistemas de tempo real e de (ultra-) alta resolução, é necessário otimizar algoritmos e sistemas. Por um lado, são desejáveis algoritmos capazes de prover alta eficiência de codificação com esforço computacional reduzido. Por outro, é necessário projetar sistemas computacionais capazes de implementar tais algoritmos de forma eficiente, reduzindo tempo de processamento, potência e, conseqüentemente, consumo de energia.

O *Versatile Video Coding* (VVC) (Bross et al., 2021) representa o estado da arte em codificação de vídeo e traz melhorias significativas quando comparado a seu antecessor, o *High Efficiency Video Coding* (HEVC) (Sullivan et al., 2012). Essas evoluções são perceptíveis em diversos aspectos do codificador e levam a uma redução média de 50% na taxa de bits em comparação com o HEVC para se obter uma qualidade de vídeo similar (Bross et al., 2021; Saldanha, 2021; Li; Zhang; Yang, 2022; Xie et al., 2022). Essa economia é crucial para aplicações em que a largura de banda é limitada ou o armazenamento é essencial.

Contudo, o aumento na eficiência de codificação do VVC vem com um custo. O tempo de codificação foi aumentado em cerca de 10 vezes em relação ao HEVC (Li; Zhang; Yang, 2022; Xie et al., 2022). Esse esforço computacional elevado representa um obstáculo para sistemas que necessitam de codificação em tempo real ou têm restrições de tempo de processamento.

Em codificadores de vídeo atuais, a predição representa uma etapa de elevada complexidade e demanda de recursos computacionais, sobressaindo-se especialmente a predição interquadros. Conforme estudos, exemplificados por Pakdaman *et al.* (Pakdaman et al., 2020), a predição interquadros representa a etapa de maior complexidade em codificadores VVC, destacando a Estimção de Movimento como o processo mais desafiador, contribuindo com cerca de 53% para a complexidade total da codificação. Além disso, as etapas de Transformadas e Quantização e de Codificação de Entropia também possuem papéis relevantes, representando, respectivamente, 22% e 7% da complexidade total. Essa complexidade está intrinsecamente ligada à quantidade de recursos computacionais exigidos. A eficiência na execução desses processos é crucial, dado o impacto significativo que eles têm no desempenho geral dos sistemas de codificação de vídeo.

Para selecionar a melhor forma de predição é preciso encontrar, dentre inúmeras opções, aquela que reduz uma função custo calculada considerando alguma métrica de distorção entre o bloco original e a predição candidata. Diversas métricas de distorção podem ser empregadas nas etapas de predição, entre elas, a soma de diferenças absolutas (*Sum of Absolute Differences* - SAD), a soma de diferenças absolutas trans-



formadas (*Sum of Absolute Transformed Differences* - SATD) e a soma de erros quadráticos (*Sum of Squared Errors* - SSE). O SAD é frequentemente adotado em sistemas de tempo real, devido à sua implementação simples, que envolve apenas somas e subtrações. Essa característica também facilita implementações em hardware visando alto desempenho e baixo consumo de energia. Adicionalmente, a organização em estrutura de árvore adotada para implementar o cálculo do SAD (como será visto ao longo desta tese) também é utilizada na implementação de SATD e SSE. Assim, a otimização de operadores para o cálculo da métrica SAD se torna um campo de pesquisa vital ao projeto de codificadores eficientes de baixo consumo de energia. Por se tratar de operadores intensamente utilizados, melhorias neste ponto dos codificadores podem oferecer melhorias ao processo de codificação como um todo.

A Computação Aproximada é explorada em vários nichos de aplicação como um meio de aumentar o a capacidade de processamento e/ou reduzir o consumo de energia em sistemas computacionais, em troca de perdas controladas de qualidade (Barua; Mondal, 2019). Esta abordagem é particularmente relevante para aplicações multimídia que naturalmente toleram imprecisões explorando limitações da visão humana sem comprometer significativamente a qualidade subjetiva do vídeo (Barua; Mondal, 2019; Mahdiani et al., 2010). Portanto, a computação aproximada representa uma alternativa promissora no projeto de codificadores de vídeo de baixo consumo de energia.

Ao tratar de operadores aritméticos aproximados, a literatura acadêmica tem se concentrado majoritariamente em somadores, destacando-se GeAr (Shafique et al., 2015), LOA (Mahdiani et al., 2010), ETA-I (Zhu et al., 2010), Copy (Gupta et al., 2013), ACAA (Kahng; Kang, 2012) e 2PSA (Porto et al., 2020). Esse enfoque é justificado pelo papel crucial que os somadores desempenham em diversas aplicações computacionais. No entanto, observa-se uma notável lacuna no que tange o estudo de subtratores. Embora seja comum a implementação de subtratores por meio de somadores, essa prática nem sempre se mostra a mais eficiente. Esse aspecto é particularmente relevante em cenários que exigem subtrações frequentes, como é o caso das métricas de distorção em codificadores de vídeo. A necessidade de uma abordagem mais eficaz para os subtratores, portanto, torna-se evidente, especialmente em contextos onde as subtrações são uma operação central e recorrente.

Portanto, a escassa discussão na literatura com foco em subtratores, aponta para uma oportunidade de pesquisa. O desenvolvimento de subtratores aproximados otimizados poderia trazer benefícios significativos para aplicações como codificadores de vídeo, onde o cálculo de métricas de distorção é essencial. Esses subtratores melhorariam a eficiência energética e de processamento, mantendo a precisão necessária para essas aplicações.

## 1.1 Tese

Com base nas avaliações já apresentadas, a tese defendida nesta pesquisa é:

*A implementação de subtratores aproximados dedicados é uma estratégia eficiente para minimizar o custo energético associado à etapa de predição nos codificadores de vídeo atuais.*

Essa tese ficará demonstrada ao final deste texto.

## 1.2 Questões de Pesquisa

Para demonstrar a validade da tese apresentada acima e melhor investigar as lacunas de pesquisa identificadas, foi formulado um conjunto de questões de pesquisa que este doutorado pretende responder.

**Questão de Pesquisa 1 (Q1):** É possível desenvolver subtratores aproximados energeticamente eficientes para aplicação na codificação de vídeo do codec VVC? Esta questão visa investigar a viabilidade de projetar subtratores que, ao mesmo tempo, maximizem a eficiência energética e mantenham uma precisão computacional adequada. Esse equilíbrio é crucial para otimizar os processos envolvidos na codificação de vídeo.

**Questão de Pesquisa 2 (Q2):** A operação de Soma das Diferenças Absolutas (SAD), utilizada nas etapas de predição, tolera bem as aproximações no codificador VVC? Esta questão de pesquisa explora a capacidade da operação SAD de aceitar aproximações sem comprometer significativamente a qualidade da predição. Essencial para os processos de predição nos codificadores de vídeo, a SAD, se for suficientemente tolerante à aproximação, pode permitir implementações que melhorem notavelmente a eficiência energética.

**Questão de Pesquisa 3 (Q3):** É viável desenvolver um sistema de hardware dedicado para predição interquadros no codec VVC que utilize subtratores aproximados nas operações de SAD? Esta questão de pesquisa explora a possibilidade de integrar subtratores aproximados em um sistema de hardware específico para o padrão VVC, investigando se tais modificações podem contribuir para uma maior eficiência sem comprometer o desempenho da predição.

## 1.3 Organização do Texto

Esta tese se dedica ao estudo de inovações na codificação de vídeo, focando no equilíbrio entre eficiência energética e precisão, e está estruturada em sete capítulos. O segundo capítulo, A Codificação de Vídeo, explora o processo de codificação, com ênfase no padrão *Versatile Video Coding* (VVC) e suas ferramentas. Computação Aproximada, o terceiro capítulo, introduz o conceito de computação aproximada e sua

aplicação em operadores matemáticos na codificação de vídeo.

O quarto capítulo, *Approximate Subtractor: Um Novo Subtrator*, apresenta subtrator aproximado proposto APPS, analisando seus resultados em comparação com outros subtratores e seu impacto na codificação de vídeo. Esse capítulo é seguido por *Análise do Algoritmo TZS com Subtratores aproximados (Test Zone Search)*, o quinto capítulo, que avalia a eficiência do TZS, focando no tamanho dos blocos e a influência dos subtratores aproximados. O sexto capítulo, *Arquitetura do TZS*, detalha a arquitetura otimizada do TZS utilizando os subtratores aproximados, incluindo metodologia de síntese e descrição de módulos relevantes.

O sétimo capítulo é a *Conclusão*, resumindo as descobertas principais e sugerindo direções futuras para a pesquisa em codificação de vídeo.

Cada capítulo contribui significativamente para a compreensão da eficiência energética e a aplicação de subtratores aproximados na codificação de vídeo, oferecendo uma base sólida para futuras pesquisas e desenvolvimentos na área.

## 2 A CODIFICAÇÃO DE VÍDEO

A compressão de vídeos digitais se tornou mandatória devido à imensa quantidade de dados necessários para representar um vídeo digital. Os vídeos são formados por sequências de quadros, que são matrizes bidimensionais de pixels. Cada pixel é representado por amostras pertencentes a canais de cores que, acumuladas em um vídeo inteiro, resultam em um volume significativo de dados (Hivo, 2024). Com o avanço para formatos de alta resolução, UHD 4K e, mais recentemente, 8K, a quantidade de dados necessária para representar um vídeo aumentou agressivamente (Lumenci, 2024).

A transmissão de vídeos, especialmente em redes móveis, enfrenta limitações de largura de banda. Apesar dos avanços tecnológicos, como a transição do 4G para o 5G, a largura de banda disponível permanece um recurso limitado. Assim, para transmitir vídeos de alta resolução e qualidade em redes móveis, como a 5G, é imprescindível a compressão do vídeo. Segundo Andrews et al. (2014), a 5G oferece uma capacidade de transmissão superior à 4G, mas a transmissão de vídeos sem compressão ainda representa um desafio significativo. A compressão de vídeos não só viabiliza a transmissão de alta qualidade em redes limitadas, mas também contribui para a economia de espaço de armazenamento.

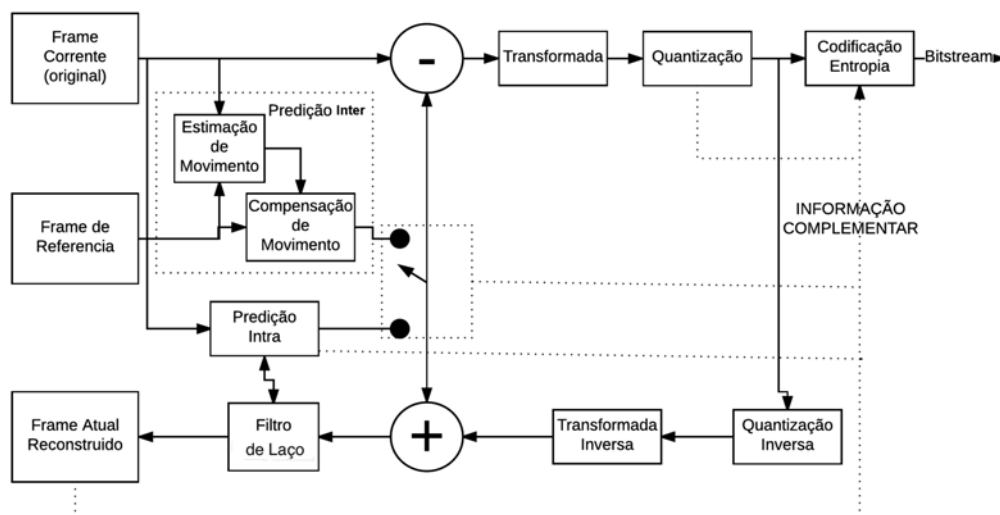
Para assegurar uma compreensão completa das seções que se seguem neste trabalho, é essencial familiarizar-se com os conceitos fundamentais delineados neste segmento inicial. A presente seção introduz uma série de tópicos essenciais de forma estruturada e sequencial. Inicialmente, discute-se sobre o Codificador de Vídeo Genérico, estabelecendo a base para os conceitos que serão abordados posteriormente. A seção que segue trata do VVC (*Versatile Video Coding*), o estado da arte dos codificadores de vídeo. No contexto do VVC, são detalhados tópicos como Estrutura de Particionamento, Quantização, Transformadas, Entropia e Filtros de Laço no VVC. Cada um desses aspectos é importante para uma compreensão mais aprofundada do processo de codificação de vídeo. O capítulo continua com a apresentação da Estimação de Movimento e de métricas de distorção, seguida de uma discussão detalhada do algoritmo de busca por zona de teste (*Test Zone Search* - TZS), uma vez que este

algoritmo é foco deste trabalho.

## 2.1 O Codificador de Vídeo Genérico

Os componentes principais de um codificador de vídeo genérico são apresentados na Figura 1.

Figura 1 – Codificador de Vídeo Genérico (Tanou; Blestel, 2019)



Uma explicação breve de cada componente principal é apresentada abaixo:

- **Predição:** A predição na codificação de vídeo serve para atenuar redundâncias tanto temporais quanto espaciais. A predição intra-quadro tem como objetivo a diminuição da redundância espacial, utilizando-se de informações de pixels adjacentes no mesmo quadro. Em contrapartida, a predição interquadros visa a redução da redundância temporal, explorando as semelhanças entre blocos dos quadros atuais e de quadros de referência temporalmente próximos. A predição interquadros se baseia em métodos como compensação de movimento (*MC*) e estimação de movimento (*ME*), onde um vetor de movimento é elaborado para identificar a localização do bloco no quadro de referência (Shi; Sun, 2008). No âmbito da compressão de vídeos, a predição assume um papel destaque, atuando na estimação estatística de variáveis aleatórias futuras com base em variáveis observáveis passadas e presentes. Nesse processo, a predição de um pixel, ou de um conjunto de pixels, é efetuada a partir dos dados de pixels já codificados. Segue-se a essa etapa a subtração dos pixels previstos dos originais, resultando em um sinal residual. A eficácia da predição é determinada pela comparação da energia do sinal residual com a do sinal original: uma menor energia residual indica uma predição bem-sucedida, permitindo uma codificação mais

eficiente dos pixels residuais através de algoritmos de codificação de entropia, reduzindo o uso de bits.

- *Transformadas e Quantização*: O módulo de transformadas é um componente importante em algoritmos de compressão de imagens e vídeos (Shi; Sun, 2008), responsável por converter dados espaciais ou temporais em uma representação de frequência. Essa conversão é fundamental para a eficiência da compressão, pois prioriza componentes relevantes do sinal, como os pixels de uma imagem. Os cálculos matemáticos envolvidos nesse processo geram um conjunto de coeficientes que representam as amplitudes das diferentes frequências presentes no sinal original. Esses coeficientes são essenciais para a compressão, uma vez que muitos deles possuem valores baixos ou nulos, o que facilita a redução do tamanho dos dados originais. Por outro lado, os coeficientes de alta frequência são menos relevantes ao sistema visual humano, podendo ser descartados ou atenuados.

A quantização, uma etapa subsequente à transformação, tem como objetivo eliminar ou atenuar os coeficientes de transformadas menos relevantes ao sistema visual humano. Essa operação atua nos coeficientes da transformada, diminuindo sua precisão através da divisão por um valor específico e subsequente arredondamento. O resultado é tipicamente uma matriz esparsa. Por outro lado, a operação de quantização implica em perda de informação. A extensão dessa perda é proporcional ao QP (*Quantization Parameter*) aplicado. Valores mais altos de QP implicam em maior perda de qualidade, mas permitem a geração de arquivos menores, enquanto valores menores preservam mais detalhes, resultando em arquivos maiores.

A escolha do nível de quantização é, portanto (Pakdaman et al., 2019; Chen; Ye; Kim, 2020), um balanço entre a qualidade desejada e o tamanho do arquivo. Em CODECs de vídeo, como VVC, a quantização é uma parte crítica do processo de compressão, com diferentes níveis aplicados conforme a importância e o conteúdo de cada parte do vídeo.

- *Codificação de Entropia*: A codificação de entropia explora a distribuição estatística dos símbolos para gerar uma representação mais compacta e sem perdas de informação, facilitando a transmissão e o armazenamento (Shi; Sun, 2008; Agostini, 2007). Nesse cenário, a distribuição estatísticas dos símbolos presentes nas matrizes esparsas geradas pela Quantização são ideais para que a codificação de entropia seja muito eficiente. O propósito principal dos codificadores que empregam a codificação de entropia é a transmissão de uma quantidade máxima de informação por símbolo codificado. Isso é alcançado representando um volume maior de informação com menos bits. Para atingir essa meta, a codificação

de entropia se vale de uma variedade de técnicas e algoritmos de compressão sem perdas. Esses métodos buscam eliminar redundâncias e aumentar a eficiência na representação dos dados. Estratégias como códigos de comprimento variável, codificação por comprimento de sequência ininterrupta de símbolos e codificação aritmética são empregadas. Essas técnicas exploram a distribuição estatística dos símbolos para criar uma representação mais enxuta e sem perda de informação, otimizando, assim, a transmissão e o armazenamento dos dados.

É importante destacar que o codificador utiliza blocos e quadros já codificados e reconstruídos para a predição, empregando informações já quantizadas e com perdas, para assegurar a compatibilidade com o processo de decodificação. Ao final do processo de reconstrução, filtros em laço, como o filtro de deblocação (*Deblocking Filter*), são aplicados para atenuar artefatos de codificação como efeitos de bloco ou *ringing*. A codificação é realizada para cada bloco de amostras de um vídeo. A principal razão para o uso de particionamento baseado em blocos na codificação de vídeo ou de imagens é a possibilidade de codificar cada bloco com uma configuração específica, selecionada de um conjunto de parâmetros predefinidos, já que um único modelo geralmente não consegue mapear eficientemente as propriedades de uma imagem completa. Ao longo do desenvolvimento das gerações de codificadores de vídeo, observa-se uma tendência de melhoria contínua nessas ferramentas. Com cada nova geração, as eficiências de codificação aumentam significativamente, muitas vezes alcançando melhorias de até 50% em comparação com as ferramentas dos padrões de vídeo anteriores. Essa evolução é bem documentada na literatura, como demonstrado por Zhang *et al.* (Zhang; Mao, 2019), refletindo os avanços tecnológicos e as necessidades crescentes por eficiência em transmissão e armazenamento de vídeo em uma era digital em expansão

A codificação de vídeo enfrenta desafios contínuos, principalmente relacionados à crescente demanda por resoluções mais altas e taxas de quadros mais elevadas. A evolução para formatos como UHD 8K exige algoritmos de compressão ainda mais eficientes. Além disso, a crescente popularização do conteúdo de Realidade Virtual (*Virtual Reality* - VR) e Realidade Aumentada (*Augmented Reality* - AR) traz novos desafios para a codificação de vídeo, devido às suas exigências únicas de formato e imersão.

## 2.2 Padrão *Versatile Video Coding* - VVC

O VVC foi padronizado pela ITU-T, ISO e IEC como Norma Internacional 23090-3 (Hhi, 2024) (MPEG-I Parte 3) (Itu, 2020), sucedendo o *High-Efficiency Video Coding* (HEVC). Ele se destaca pela redução significativa na taxa de bits em relação ao seu antecessor HEVC, o que é decisivo para o *streaming* de vídeos em altas resoluções.

Esse desenvolvimento atende a demanda crescente por vídeos de alta qualidade, impulsionada pela evolução tecnológica das TVs domésticas e pela expansão do acesso à internet de alta velocidade.

O novo padrão mantém a tradicional codificação de vídeo híbrida baseada em blocos, e sua interface de sistema e transporte segue a estrutura de fluxo de bits baseada em unidades de camada de abstração de rede (*Network Abstraction Layer* - NAL) e conjuntos de parâmetros, semelhante aos padrões anteriores (Springerlink, 2024). Essa continuidade facilita a integração do VVC em diferentes plataformas e formatos, incluindo *streaming*, transmissão e videoconferência, compatível com formatos de arquivo de mídia base ISO (ISO/BMFF) (al., 2024a).

Entre as principais inovações do VVC, destaca-se o particionamento de bloco versátil, que melhora significativamente o processamento de diferentes conteúdos de vídeo (al., 2024a). A adição de modos angulares para a predição intra-quadro otimiza a eficiência na análise de mudanças dentro de um quadro único, enquanto a capacidade de compensação de movimento aprimora o tratamento de movimentos complexos no vídeo.

Outro novo recurso do VVC é a seleção múltipla de transformadas, que permite escolhas mais efetivas entre diferentes métodos de transformação, adaptando-se com precisão ao conteúdo do vídeo. A implementação de Precisão Adaptativa dos Vetores de Movimento (*Adaptive Motion Vector Resolution* - AMVR) é um progresso considerável, possibilitando a codificação de vetores de movimento com variados níveis de precisão (al., 2024b).

Além disso, o VVC introduz avanços na Estimção de Movimento Fracionária (*Fractional Motion Estimation* - FME), com uma abordagem que inclui o uso de um filtro Gaussiano alternativo de seis pontos para deslocamentos de 1/2 pixel, diferenciando-se do HEVC em sua metodologia. Além de atender a uma ampla gama de formatos, incluindo vídeos de faixa dinâmica padrão (*Standard Dynamic Range* - SDR), de faixa dinâmica elevada (*High Dynamic Range* - HDR), de ampla gama de cores (*Wide Color Gamut* - WCG), de conteúdo de tela (*Screen Content* - SC) e vídeos de 360 graus, o VVC inova no tratamento de informações de usabilidade de vídeo (*Video Usability Information* - VUI) e informações suplementares sobre melhorias (*Supplemental Enhancement Information* - SEI). As mensagens SEI que influenciam a conformidade do fluxo de bits são integradas ao padrão, enquanto os parâmetros de VUI e outras mensagens SEI são especificadas separadamente no padrão *Versatile SEI* (VSEI) (al., 2024c).

Com suas inovações, o VVC alcança uma eficácia até 50% superior ao HEVC (Bross et al., 2020), atendendo às crescentes demandas por conteúdo de vídeo de alta qualidade.



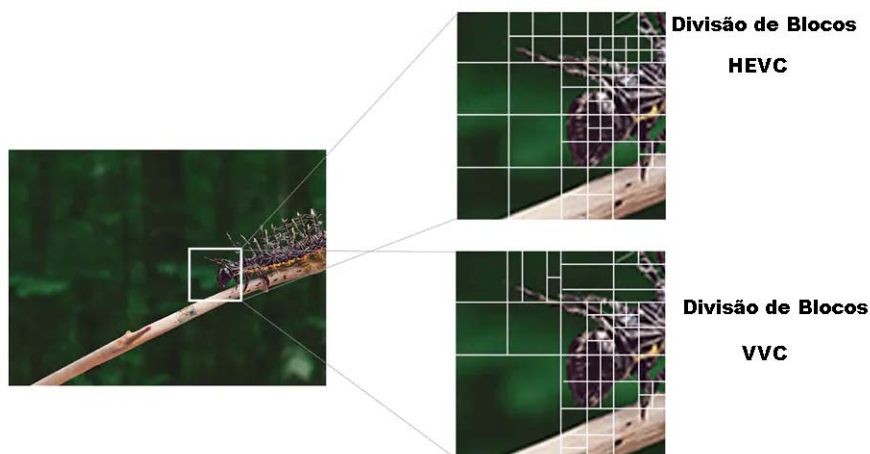
### 2.2.1 Estrutura de Particionamento

O VVC representa um salto evolutivo na codificação de vídeo, superando o HEVC. Esse avanço se manifesta principalmente na eficiência de compressão e na precisão do particionamento dos quadros, conforme Figura 2.

Inicialmente, o VVC divide os quadros em *Coding Tree Units* (CTUs). Uma CTU é uma unidade que abrange uma região do vídeo a ser codificada. No VVC, as CTUs podem chegar até 128x128 pixels, permitindo abranger áreas maiores em comparação ao HEVC, que tem um limite de 64x64 pixels. Essa característica é importante para lidar com resoluções mais altas, como HD e UHD 4K, favorecendo a codificação de áreas homogêneas. Contudo, isso resulta em um aumento do custo computacional (Yoon; Kim, 2022).

Após a divisão em CTUs, o processo segue para a subdivisão em *Coding Units* (CUs) através de uma estrutura de árvore quadrática (Quadtree - QT). A QT é uma forma de representar a CTU em blocos que podem ser divididos até o tamanho mínimo de 4x4 pixels. Essa subdivisão é essencial para adaptar a codificação à complexidade do conteúdo do vídeo, otimizando a precisão, especialmente em áreas com detalhes finos.

Figura 2 – Divisão de Blocos no HEVC e VVC (Sjöberg et al., 2020)



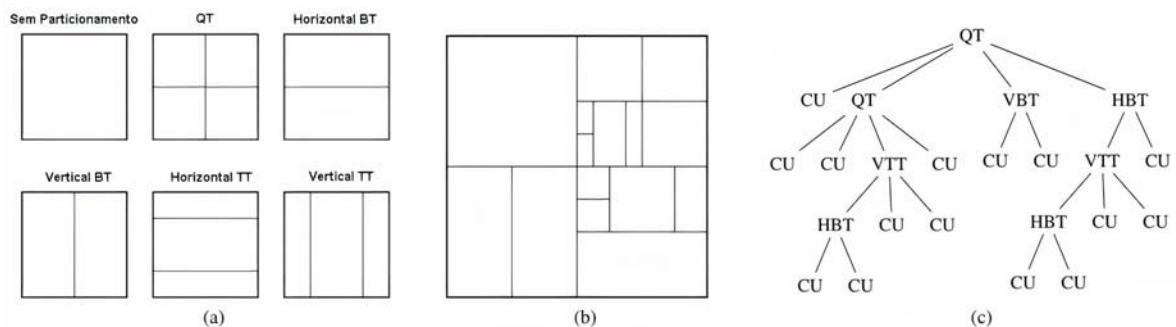
O VVC introduz uma nova estrutura de particionamento, árvore multi-tipo (*Multi-Type Tree* - MTT), que confere maior flexibilidade ao processo. A MTT é aplicada a cada uma das folhas da QT e possibilita divisões binárias (*Binary Tree* - BT) e ternárias (*Ternary Tree* - TT). A BT divide um bloco em duas partes simétricas, enquanto a TT o divide o bloco em três partes na proporção 1:2:1, como pode ser visto na Figura 3. Esse mecanismo, aplicável tanto horizontal quanto verticalmente, permite uma adaptação mais apurada ao conteúdo do vídeo.

A união da QT com a MTT gera a árvore quarternária com árvores multi-tipo aninhadas (*Quadtree with nested Multi-Type Tree* - QTMT) que é a árvore final de partici-

onamento do VVC. Cada folha da QTMT é uma CU e esse particionamento é usado tanto como unidades de predição quanto de transformação. Para cada nível, a QTMT possibilita seis tipos de partições de CU, conforme apresentado na Figura 3, oferecendo assim uma flexibilidade notável na representação de tamanhos e formas de blocos.

Cada CU é avaliada pelo processo de otimização taxa-distorção (*Rate-Distortion Optimization* - RDO) para determinar a partição mais eficaz, considerando todas as possíveis divisões de estrutura.

Figura 3 – Esquema de Particionamento de Blocos no VVC (Gao et al., 2020)



### 2.2.2 Predição Intra-Quadro

A predição intra-quadro é uma ferramenta chave na codificação de vídeo, especialmente no padrão VVC. Essa ferramenta visa prever o conteúdo de blocos individuais de um quadro, utilizando informações previamente codificadas no mesmo quadro. O principal objetivo é a redução da redundância e do volume de dados necessários para a representação eficiente do quadro.

O procedimento inicia com a seleção de blocos, onde o bloco é segmentado em unidades menores de codificação. Conforme ilustrado na Figura 4, amostras acima e à esquerda do bloco, já codificadas, são empregadas como referências nesse processo. Posteriormente, são aplicados diversos modos de predição, tais como a predição angular, DC e planar, para estimar o conteúdo de cada bloco. No contexto do VVC, modos mais avançados, incluindo a predição baseada em matriz e a predição de múltiplas linhas de referência, são igualmente utilizados.

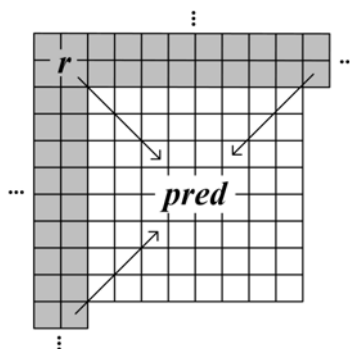
O codificador desempenha um papel essencial na seleção do modo de predição mais eficiente, buscando minimizar a diferença entre o conteúdo original e o previsto do bloco (Morin, 2020). Essa diferença, geralmente quantificada pela soma dos erros quadráticos, é conhecida como resíduo. O resíduo é, então, codificado e transmitido, acompanhado das informações relativas ao modo de predição escolhido. Em situações onde a predição é extremamente eficaz, o resíduo pode ser insignificante ou até mesmo nulo.

O HEVC implementa 35 modos de predição intra, incluindo o modo planar. Por sua

vez, o VVC expande essa gama para 65 modos, como mostrado na Figura 5, incluindo modos angulares direcionais. São utilizados dois conjuntos de filtros de interpolação de quatro pontos com diferentes cortes de frequência para gerar amostras de predição em posições fracionárias nos modos angulares. O codificador estende a filtragem para incluir  $45^\circ$  na diagonal e modos adjacentes na horizontal e na vertical, empregando a técnica de combinação de previsão dependente de posição (*Position-Dependent Prediction Combination* - PDPC), para aprimorar a qualidade da predição no modo planar.

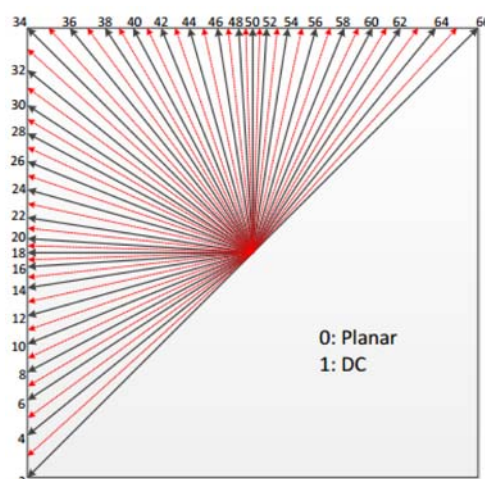
A predição intra-quadro do VVC aprimora as amostras de referência, possibilitando o uso de múltiplas linhas de referência. Segundo Pfaff (Pfaff et al., 2021), a predição intra com múltiplas linhas de referência (*Multiple Reference Line Intra Prediction* - MRL) melhora significativamente a qualidade da predição. O VVC introduz três novas abordagens para a predição de blocos intra: o modo *Cross Component Linear Model* (CCLM), onde as amostras de croma são previstas a partir das amostras de luminância reconstruídas da mesma CU, utilizando um modelo linear; as sub-partições intra (*Intra Sub-Partitions* - ISP), dividindo o bloco vertical ou horizontalmente em duas ou quatro subpartições; e a predição intra baseada em matriz (*Matrix-based Intra Prediction* - MIP), que utiliza uma linha de amostras vizinhas reconstruídas dos blocos à esquerda e acima como vetores de entrada, realizando, após um pré-processamento, a interpolação linear nas direções vertical e horizontal.

Figura 4 – Exemplificação da Intra-predição em um Bloco. (Pfaff et al., 2020)



Diversas outras ferramentas de codificação são empregadas no processo de predição intra para aprimorar o desempenho da codificação, aumentando assim a complexidade temporal. O método modelo linear de componentes cruzados (*Cross-Component Linear Model* - CCLM) assume que valores de amostras de luminância reconstruídos podem auxiliar na predição das amostras de croma através de um modelo linear, tornando a compressão das amostras de croma mais eficiente. Devido às formas variáveis de CU geradas pela MTT, o VVC introduz 28 modos de AIP de ângulo largo (*Wide Angular Intra Prediction* - WAIP) para predição angular assimétrica.

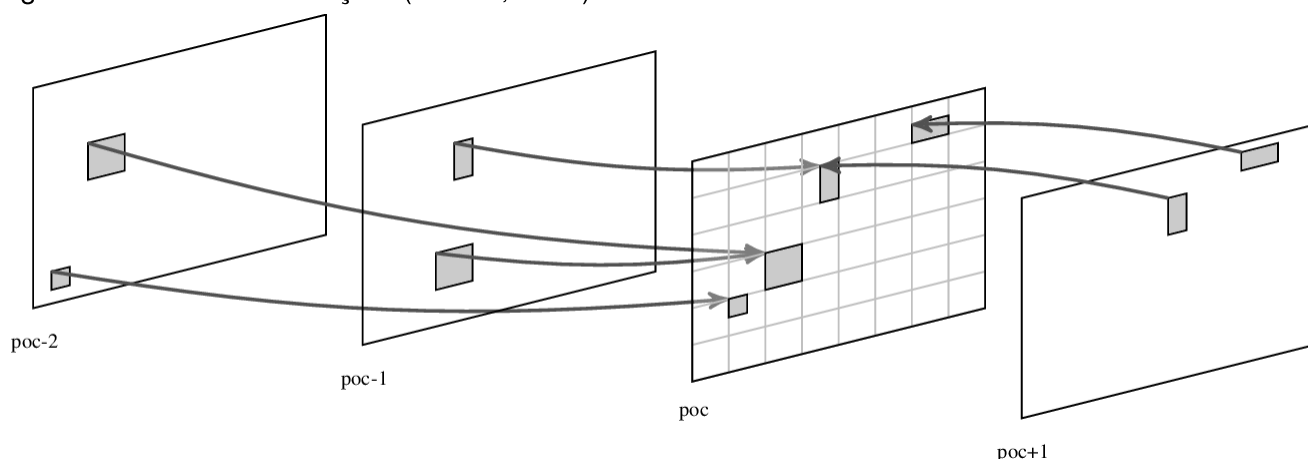
Figura 5 – Modos de Predição Intra. (Pfaff et al., 2020)



### 2.2.3 Predição Inter-Quadros

A técnica de codificação inter-quadros desempenha um papel primordial na exploração e minimização da redundância temporal entre quadros consecutivos em vídeos, buscando as similaridades entre cada CU do quadro atual e as CUs previamente codificadas em outros quadros, chamados quadros de referência. Ela opera através da geração de um vetor de movimento (VM), que determina, dentro dos quadros de referência, a posição relativa da CU mais similar à CU atual, como exemplificado na Figura 6. Ao reduzir a correlação temporal entre as imagens, essa técnica contribui significativamente para a eficiência da codificação e para uma compressão de vídeo de alta qualidade.

Figura 6 – Blocos de Predição. (Correa, 2014)



Dentre os avanços nesse campo no VVC, destaca-se a FME, que inova ao buscar movimentos com precisão fracionária, iniciando com meio pixel e avançando para quarto de pixel, a partir de uma estimativa de pixel inteiro. Tal atenção aos detalhes nos deslocamentos fracionários de pixels assegura vetores de movimento mais precisos, elevando a eficiência do processo de codificação.

No âmbito do VVC, ressaltam-se as inovações na estimação de movimento, como a Predição de MV Baseada em História (*History-Based MV Prediction* - HMVP) e a Resolução Adaptativa de MV (*Adaptive MV Resolution* - AMVR). A HMVP otimiza a predição ao permitir a reutilização de MVs de CUs previamente codificadas, enquanto a AMVR introduz uma resolução flexível para os MVs, equilibrando a precisão com a carga de bits necessária.

O VVC também avança na estratégia de predição inter-quadros, empregando informações de quadros anteriores ou posteriores para codificar blocos específicos. Essa abordagem é fortalecida por técnicas como HMVP e Codificação de Vetor de Movimento, que maximizam a eficiência da codificação.

Além disso, o VVC incorpora modos inovadores, como o Modo de Fusão com Diferença de Vetor de Movimento (*Merge Mode on Motion Vector Difference* - MMVD), a Predição Combinada Inter-Intra (*Combined Inter-Intra Prediction* - CIIP) e o Modo de Partição Geométrica (*Geometric Partitioning Mode* - GPM).

A Estimação de Movimento Afim (*Affine Motion Estimation* - AME) representa outra inovação significativa, empregando um modelo afim para descrever transformações complexas, como rotação, escala e cisalhamento, além da translação. Esse modelo, mais abrangente do que um simples vetor de movimento, é descrito por uma matriz de transformação que contém uma gama diversificada de parâmetros.

As inovações da predição inter-quadros do VVC representam uma evolução significativa em relação aos seus antecessores, como o HEVC, introduzindo novas ferramentas e técnicas que resultam em melhorias notáveis na eficiência de codificação, apesar do aumento do custo computacional do codificador.

#### 2.2.4 Transformadas

Segundo os autores Chen (Chen; Ye; Kim, 2020), Hernandez (De-Luxán-Hernández et al., 2019) e Zhao (Zhao et al., 2021), as técnicas avançadas de codificação de transformada utilizadas no padrão VVC são cruciais para aprimorar a eficiência da codificação. O principal objetivo dessas técnicas é alcançar uma compressão mais eficaz dos dados de vídeo.

No VVC, são introduzidas quatro novas técnicas de codificação de transformadas, todas contribuindo para ganhos significativos na eficiência de codificação. Essas técnicas são: (i) seleção múltipla de transformada (*Multiple Transform Selection* - MTS), que permite a seleção de diferentes transformadas para otimizar a relação entre taxa e distorção, incluindo a tradicional transformada discreta dos cossenos do tipo II (DCT II), mas também a DCT VIII e a transformada discreta dos senos do tipo VII (DST VII), onde diferentes combinações na horizontal e na vertical são permitidas; (ii) transformada secundárias não separável de baixa frequência (*Low-Frequency Non-separable Secondary Transform* - LFNST), uma transformada secundária que atua sobre coefici-

entes de baixa frequência oriundos da predição intra-quadros e (iii) Transformada de sub bloco (*Sub-Block Transform* - SBT), usada para lidar com amostras residuais em pequenas regiões.

A comparação entre as transformadas do VVC e do HEVC apresenta diferenças significativas. O VVC apresenta uma maior variedade de transformadas, como MTS e LFNST, que não estão presentes no HEVC. Além disso, as transformadas no VVC são mais complexas, mas oferecem maior eficiência na codificação, sendo particularmente efetivas para resoluções mais altas e conteúdos detalhados. O VVC também introduz transformadas de tamanhos maiores e tipos diferentes, como DST VII e DCT VIII, que são mais adaptáveis a variados tipos de conteúdo de vídeo, em contraste com o HEVC. Além disso, o VVC suporta transformadas de tamanhos maiores, até 64x64, enquanto o máximo no HEVC era 32x32. O VVC também suporta transformadas retangulares, algo não previsto no HEVC.

### 2.2.5 Quantização

As inovações na quantização no VVC, particularmente a quantização dependente escalar, destacam-se em relação aos padrões anteriores, como o HEVC. Essa abordagem inovadora é detalhada em estudos recentes, como os de Pakdaman (Pakdaman et al., 2019), Chen (Chen; Ye; Kim, 2020) e (Zhao et al., 2021). Nessa metodologia, a seleção dos níveis de quantização em um bloco é otimizada com base nos valores dos coeficientes já quantizados, uma estratégia que equilibra a contagem de bits e o erro de reconstrução total do bloco. Essa técnica resulta em uma significativa redução de bits, superando a eficiência do HEVC ao eliminar a ineficiência inerente à quantização independente de coeficientes.

A diferença notável entre a quantização no HEVC e no VVC reside na complexidade e na eficiência dos métodos empregados. O VVC adota técnicas de quantização mais sofisticadas, como a Quantização Trellis-Codificada (TCQ), melhorando a eficiência de compressão e oferecendo qualidade de vídeo superior mesmo a taxas de bits reduzidas. Esse avanço é particularmente relevante para conteúdos de vídeo de alta resolução e qualidade, onde a relação entre taxa e distorção é um fator crítico.

Entre as inovações no padrão VVC, a Quantização Trellis-Codificada (TCQ) e os Quantizadores de Reconstrução Uniforme (URQs) são particularmente importantes. Ambas as técnicas são aprimoradas no VVC para proporcionar uma codificação mais eficiente em comparação com os métodos tradicionais. A introdução do TCQ, em particular, marca uma distinção significativa em relação ao HEVC, permitindo uma quantização mais eficiente dos coeficientes de transformada e, conseqüentemente, uma compressão mais eficaz, preservando uma alta qualidade visual, mesmo em taxas de compressão elevadas.

### 2.2.6 Codificação de Entropia

O padrão VVC apresenta um avanço significativo na codificação de entropia, superando o HEVC. Ambos os padrões empregam a codificação de entropia binária adaptativa ao contexto (*Context-Adaptive Binary Arithmetic Coding* - CABAC), um método eficiente para representar informações com um número reduzido de bits, preservando a integridade dos dados. No VVC foram implementadas melhorias no CABAC, incluindo a codificação de coeficientes aprimorada, que otimiza a codificação do HEVC, e a varredura de coeficientes em diagonal reversa, melhorando o processamento e a codificação dos coeficientes em blocos de transformadas.

Segundo pesquisadores, como Park et al. (Park et al., 2019) e Karwowski et al. (Karwowski, 2021), o VVC aprimorou a seleção do modelo de probabilidade no CABAC. Isso permite uma compressão mais eficiente ao selecionar modelos de probabilidade mais precisos para níveis absolutos de coeficientes de transformação, fundamental para alcançar altas taxas de compressão sem comprometer a qualidade do vídeo.

O VVC também introduz inovações no CABAC, como a modelagem de probabilidade de Bins com alta granularidade e a estimação de probabilidade de Bin com múltiplos parâmetros.

Essas inovações demonstram o avanço na compressão de vídeo, onde a precisão na modelagem de entropia e a adaptação aos dados do vídeo são essenciais. O VVC CABAC supera tecnologias anteriores em termos de mecanismos de estimação de probabilidade dos bins e modelagem estatística aprimorada, refletindo a evolução contínua na área de codificação de vídeo.

### 2.2.7 Filtros de Laço

O VVC integra três tipos principais de filtros de laço, nomeadamente. Esses filtros compreendem o filtro de deblocação (*Deblocking Filter* - DBF), o filtro de amostra de deslocamento adaptativo (*Sample Adaptive Offset* - SAO), e uma inovação do VVC, o filtro de laço adaptativo (*Adaptive Loop Filter* - ALF) (Sjöberg et al., 2020), (Bross et al., 2021). O ALF é particularmente notável pela sua habilidade em aplicar seletivamente uma variedade de filtros FIR bidimensionais pós-decodificação, com base em uma classificação de bloco que possibilita a seleção dentre 25 filtros diferentes. Essa característica é essencial na minimização de artefatos e na otimização da compressão de dados.

O método de filtragem em blocos do VVC (Karczewicz et al., 2021), essencial para mitigar artefatos da abordagem de codificação baseada em blocos, é aprimorado através do uso de filtros de deblocação mais extensos. Eles são particularmente eficazes na redução de artefatos em blocos maiores, contribuindo significativamente para a melhoria da qualidade visual dos vídeos codificados pelo VVC em comparação ao

HEVC.

Adicionalmente, o codificador apresenta o filtro de deblocação de luminância adaptativo, um filtro que ajusta a intensidade de deblocação com base no nível médio dos valores de amostra de luminância reconstruídos. Esse filtro se mostra extremamente útil para conteúdos HDR, dada a sua característica de transferência não linear distinta. A adaptabilidade do filtro de deblocação à luminância é importante para aprimorar a qualidade em áreas de alta ou baixa luminosidade, atenuando distorções visuais.

No contexto do VVC, o ALF melhora o sinal de vídeo reconstruído através de um processo de filtragem espacial, utilizando uma região de suporte em forma de diamante para os componentes de luminância e croma. O codificador, ao selecionar e sinalizar parâmetros de *clipping* adequados, baseia-se na similaridade entre amostras adjacentes, promovendo uma adaptabilidade local mais eficiente na seleção de filtros.

A introdução do ALF entre componentes (*Cross-Component* - CC-ALF) representa outra inovação, aplicando um filtro passa-alta em forma de diamante por componente de croma. Essa abordagem oferece uma adaptabilidade local significativa, permitindo ao codificador determinar conjuntos de coeficientes de filtro específicos por componente de croma, que são transmitidos juntamente com os parâmetros do ALF.

Os filtros em laço do VVC são processados em uma sequência específica durante a reconstrução do quadro. O DF, uma evolução do filtro HEVC, é empregado para mitigar artefatos de bloco, adaptado às novas estruturas de bloco do VVC. O SAO, mantendo seu design original do HEVC, é utilizado para reduzir artefatos de *ringing*, ajustando adaptativamente a intensidade dos pixels. O ALF, reintegrado ao VVC após revisões de implementação, aumenta a eficiência da codificação ao aplicar uma filtragem linear e *clipping* adaptativo.

Antes da aplicação dos filtros em laço, o VVC emprega o mapeamento de luminância com dimensionamento de croma (*Luma Mapping with Chroma Scaling* - LMCS) para ajustar a gama dos valores de amostra de luminância, visando otimizar a qualidade tanto subjetiva quanto objetiva da sequência de vídeo codificada.

As inovações introduzidas pelos filtros de laço no codificador VVC são notáveis e representam avanços significativos em relação a seus predecessores, como o HEVC. Tais inovações incluem a melhoria na restauração do sinal pelo ALF, a correção adaptativa de cores pelo CC-ALF, a adaptação da distribuição de amostras pelo LMCS, a eficácia aprimorada do DBF na suavização de discontinuidades de bloco e as melhorias no SAO para enfrentar artefatos de quantização. Esses avanços são fundamentais para a compressão de vídeo mais eficiente, resultando em uma melhoria substancial na qualidade da imagem e na eficiência da taxa de bits em comparação com os pa-



drões anteriores de codificação de vídeo.

## 2.3 Estimação de Movimento

A estimação de movimento é um componente crítico na codificação de vídeo, envolvendo a identificação de vetores de movimento que mapeiam as redundâncias entre quadros consecutivos em um vídeo digital. Esse processo busca o bloco nos quadros de referência que melhor corresponde a um bloco no quadro atual. Os blocos candidatos dentro da área de busca são avaliados para determinar o melhor representante do bloco original e esse processo é realizado através de um algoritmo de busca. A seguir, descrevemos alguns algoritmos de busca da estimação de movimento utilizados em codificadores contemporâneos:

**Busca Exaustiva:** O algoritmo de busca exaustiva (*Full Search* - FS) se destaca por realizar uma busca em todos os blocos candidatos dentro de uma área de pesquisa predefinida. Essa técnica é essencial para identificar o vetor de movimento ótimo para aquela área de pesquisa, que corresponde ao bloco candidato com o menor resíduo. O FS, mencionado em estudos como os de Lin (2005), (Richardson, 2004; Lin; Leou, 2005; Marcelo Porto, 2012), desloca o bloco do quadro atual, amostra por amostra, aplicando um critério de distorção, como a Soma das Diferenças Absolutas (SAD), em cada bloco candidato. Essa abordagem garante a identificação do bloco com a menor diferença, como ilustrado na Figura 7.

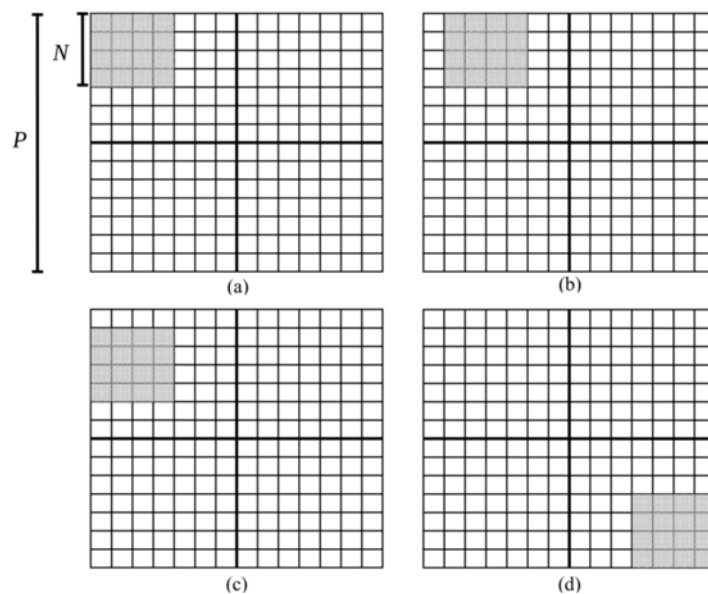
O processo de busca do FS pode assumir diferentes formas. Uma delas, como amplamente discutido pela literatura (Richardson, 2004; Lin; Leou, 2005; Marcelo Porto, 2012), é iniciar do canto superior esquerdo da área de busca, prosseguindo de maneira sequencial, linha a linha, até o canto inferior direito. Nesse contexto, a Figura 7 demonstra o algoritmo FS em ação, começando no canto superior esquerdo e movendo-se amostra por amostra. Considerando uma área de busca de  $\pm 5$  amostras ao redor de um bloco central de  $4 \times 4$  amostras, o algoritmo compara o bloco atual com cada um dos blocos candidatos, recalculando a diferença a cada novo posicionamento.

O esforço computacional do FS é elevado devido à característica de comparar todos os blocos dentro da área de busca. Esse esforço aumenta com o aumento do tamanho da área de busca, como destacado em estudos de (Porto, 2008). Embora a expansão da área de busca possa reduzir significativamente o resíduo, os ganhos em termos de eficiência de codificação não são proporcionais ao aumento da área. Essa característica torna a utilização do FS em software uma tarefa desafiadora especialmente para o processamento em tempo real de vídeos de alta resolução, demandando o desenvolvimento de soluções arquiteturais especializadas. Essas soluções, apesar de exigirem maior área e potência em comparação com algoritmos heurísticos (que serão discutidas a seguir), possibilitam a exploração do paralelismo, graças à inde-

pendência de dados do algoritmo FS, atendendo aos requisitos de desempenho para aplicações focadas em qualidade.

No que tange à aplicação prática do FS, é importante considerar as dimensões da área de busca e dos blocos. Para uma área de  $P \times P$  e blocos de  $N \times N$ , o número de blocos candidatos na área é  $(P - N + 1) \times (P - N + 1)$ . Consequentemente, para cada bloco candidato,  $N^2$  comparações de amostras são necessárias. Tomando como exemplo um quadro Full HD de resolução 1920x1080, com área de busca de 64x64 e blocos de 4x4 amostras, obtemos 129.600 blocos em um quadro e 3.721 blocos candidatos em uma área de busca. Isso resulta em aproximadamente 7.715.865.600 comparações para gerar os vetores de movimento de todos os blocos do quadro.

Figura 7 – Algoritmo de Busca Completa. (Marcelo Porto, 2012)



**Busca em Diamante:** O algoritmo de busca em diamante (*Diamond Search* - DS) é amplamente reconhecido e utilizado na literatura. Em comparação com o algoritmo FS, o DS consegue reduzir significativamente o esforço computacional sem comprometer demasiadamente a qualidade dos resultados. A estratégia do DS envolve a utilização de um padrão de busca em forma de diamante, dividido em duas fases: o Padrão de Busca de Diamante Grande (*Large Diamond Search Pattern* - LDSP) e o Padrão de Busca de Diamante Pequeno (*Small Diamond Search Pattern* - SDSP), conforme apresentado por Zhu (2000). A Figura 3.8 ilustra esses padrões, onde cada intersecção indica um bloco candidato e os círculos representam a posição dos blocos candidatos sendo avaliados. Os círculos em branco (L) indicam o padrão LDSP, e os círculos em cinza (S) o padrão SDSP.

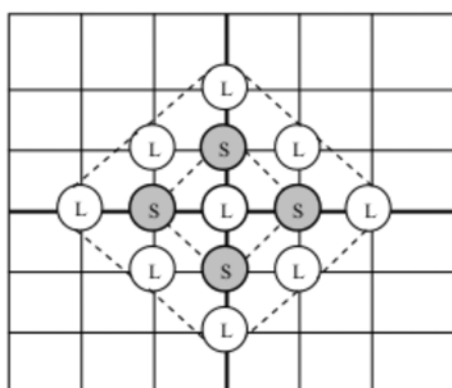
O padrão LDSP, usado na primeira fase da busca, consiste em um diamante de nove blocos candidatos ao redor do centro da área de busca. Após essa etapa, o padrão SDSP, formado por quatro blocos candidatos adjacentes à posição central do LDSP, é utilizado para refinar o resultado. Essa abordagem é descrita em detalhes

pelos autores (Ravuri, 2023; Lai; Chen; Huang, 2010; Marcelo Porto, 2012; Kuo et al., 2009) O processo inicia com a aplicação do LDSP ao centro da área de busca, compreendendo nove comparações. Essa estratégia explora a tendência dos vídeos digitais de apresentarem movimentos de blocos próximos às suas posições originais. A primeira fase termina quando o menor resíduo é encontrado no centro do LDSP.

Em cerca de 77% dos casos, o DS requer múltiplas aplicações do LDSP. Nessa situação, um novo LDSP é posicionado no local de menor resíduo encontrado anteriormente. Dependendo se este novo centro está em uma aresta ou vértice do diamante, a estratégia de busca varia, como demonstrado na Figura 8. A busca continua até que o menor erro seja localizado no centro do LDSP, momento em que o padrão SDSP é aplicado como refinamento e o bloco com a menor distorção é o escolhido.

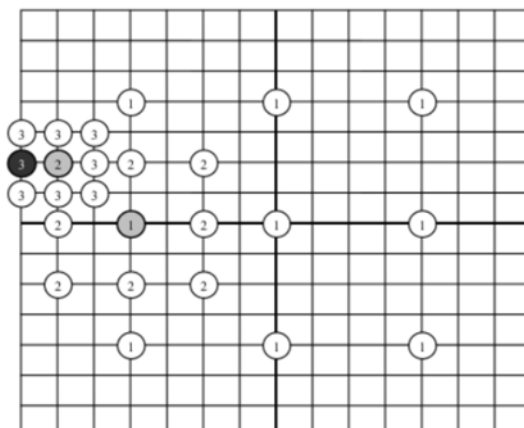
A quantidade de iterações do LDSP necessárias para alcançar o melhor resultado varia conforme as características do vídeo em processamento. Essa é uma característica pouco desejável, uma vez que, a priori, é impossível saber quantas interações serão necessárias. Em média, o DS executa cerca de três iterações para vídeos de 740x480 pixels, conforme indicado por Porto (Porto, 2008). No melhor cenário, sem nenhuma iteração, o DS avalia apenas 13 blocos candidatos para gerar um vetor de movimento. Em um caso médio, com três iterações e uma média de quatro blocos candidatos por LDSP, são analisados cerca de 25 blocos. Comparando com o exemplo anterior do FS, o DS necessitaria de 26,956,800 comparações no melhor caso e 51,840,000 no caso médio, representando uma redução significativa no número de comparações, reduzindo entre 149 e 286 vezes o número de comparações em relação ao FS.

Figura 8 – Algoritmo de Busca Diamond. (Marcelo Porto, 2012)



**Busca em Três Etapas:** O algoritmo de busca em três etapas (*Three Step Search* - TSS), oferece um método previsível na busca de vetores de movimento para estimação de movimento. Esse algoritmo, diferenciando-se do DS pela sua regularidade, possui um número fixo de comparações, o que confere uma vantagem significativa em termos de esforço computacional (Li; Zeng; Liou, 1994; Jing; Chau, 2004; Marcelo

Figura 9 – Algoritmo de busca Three Step Search. (Marcelo Porto, 2012)



Porto, 2012). Cada vetor de movimento no TSS é gerado após um número constante de comparações, garantindo uniformidade no processamento. A Figura 9 ilustra o funcionamento do TSS, demonstrando sua aplicação em uma área de busca de sete amostras.

Inicialmente, o TSS compara nove blocos candidatos no primeiro passo: o bloco central e outros oito blocos situados a uma distância  $D$  amostras ao redor do centro da área de busca (sendo  $D = 4$  no exemplo dado). Esses blocos são avaliados para determinar aquele com a menor distorção, que então se torna o novo centro para a busca subsequente. Após essa etapa, o valor de  $D$  é reduzido pela metade.

No segundo passo, o algoritmo examina mais oito blocos candidatos localizados a uma distância  $D$  do novo centro da busca, o qual foi determinado pelo melhor resultado do passo anterior. A distância  $D$  é novamente atualizada para  $D/2$ . O bloco candidato com a menor distorção é selecionado como o novo centro para o terceiro e último passo.

Nesta fase final, o TSS compara mais oito blocos candidatos em torno do centro atualizado, selecionando, ao final, o bloco com a menor distorção para a geração do vetor de movimento. Ao longo dos três passos, o TSS realiza um total de 25 comparações de blocos candidatos (nove no primeiro passo e oito em cada um dos dois passos subsequentes), estabelecendo esse o custo fixo do algoritmo para a geração de cada vetor de movimento. Em contraste com o DS, as características específicas do vídeo processado não influenciam o custo computacional do TSS, visto que o algoritmo possui sempre três etapas iterativas apenas.

### 2.3.1 Métricas de Distorção

A avaliação da distorção entre dois blocos de amostras é um aspecto indispensável na codificação de vídeo. Diversas métricas são empregadas para quantificar esse aspecto. Entre elas, a Soma das Diferenças Absolutas (SAD) e outras métricas de

distorção representam aproximadamente 40% do tempo total de execução do codificador de vídeo HEVC, conforme indicado por Bossen (2012) e Silveira (2018) (Bossen et al., 2012; Silveira et al., 2018). Nesta seção, são descritas algumas das métricas comumente utilizadas nos codificadores atuais: SAD, Soma das Diferenças Quadráticas (*Sum of Squared Errors* - SSE) e Soma das Diferenças Absolutas Transformadas (*Sum of Absolute Transformed Differences* - SATD). As métricas de distorção apontam o nível de diferença entre o bloco atual e o bloco candidato. Assim, quanto maior o valor indicado pelas métricas SAD, SSE ou SATD, maior é a distorção entre os blocos comparados ou, em outras palavras, menor é a similaridade entre eles.

- **Soma das Diferenças Absolutas (SAD):** Esta métrica é a soma dos valores absolutos das diferenças entre cada amostra do bloco original (O) e do bloco candidato (R). A letra R é utilizada para denotar o bloco candidato, pois ele pertence a um quadro reconstruído, equação 1:

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |O_{i,j} - R_{i,j}| \quad (1)$$

- **Soma das Diferenças Quadráticas (SSE):** A SSE é a soma dos quadrados das diferenças entre cada amostra do bloco original e do bloco candidato. Essa métrica é particularmente eficaz por destacar erros de maior amplitude e é definida na equação :

$$SSE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (O_{i,j} - R_{i,j})^2 \quad (2)$$

- **Soma das Diferenças Transformadas Absolutas (SATD):** A SATD, baseada na SAD, utiliza a Transformada de Hadamard (H) das diferenças absolutas de  $O_{i,j} - R_{i,j}$  antes de realizar a soma. Apesar de ter um custo computacional mais elevado que a SAD, oferece uma qualidade superior ao operar no domínio das frequências, equação 3.

$$SATD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |H_{(i,j)}| \quad (3)$$

A SAD, devido à facilidade de implementação, envolvendo apenas somas e subtrações, é amplamente adotada no desenvolvimento de hardware, como destacado por Porto (2017) (Porto et al., 2017). Portanto, a métrica SAD será adotada nos experimentos descritos ao longo desta tese.

## 2.4 Algoritmo Test Zone Search

O algoritmo de busca por zona de teste (*Test Zone Search*- TZS) é um algoritmo de busca da estimação de movimento, mas por sua importância para este trabalho por ser implementado no VVC, o TZS será detalhado neste texto. O TZS reduz o tempo de codificação do FS em mais de 60%, atingindo resultados de qualidade similares (Doan et al., 2017). Diferentemente do FS, que analisa todas as possíveis localizações de movimento dentro de um intervalo específico, para encontrar a melhor correspondência de bloco, o TZS adota uma estratégia mais refinada e otimizada. Essa técnica limita a busca a áreas específicas, identificadas como mais propensas a conter o movimento desejado, com base em uma variedade de heurísticas. Tal abordagem acelera o processo de busca tornando o TZS mais adequado que o FS para dispositivos com capacidade computacional limitada.

A superioridade do TZS em relação ao FS é evidenciada principalmente por sua eficiência de processamento e velocidade. No contexto da codificação de vídeo, "eficiência" refere-se à capacidade de realizar a estimação de movimento de forma rápida, empregando o mínimo possível de recursos computacionais, sem comprometer significativamente a precisão. O TZS, através de suas heurísticas, efetua uma busca focada e inteligente. Isso resulta em uma diminuição considerável do número de comparações necessárias. Por exemplo, em um vídeo de 720p, com uma área de busca de 64x64 e blocos de 4x4 amostras, são realizadas 214.329.600 comparações. Com o TZS, esse número pode ser reduzido pela metade, dependendo da heurística aplicada (Goncalves et al., 2019).

Essa diminuição não apenas agiliza o processo de busca, mas também alivia a carga sobre os recursos de processamento. Em aplicações que requerem alto desempenho em tempo real essa eficiência é essencial. Adicionalmente, em dispositivos com recursos limitados, como *smartphones* ou câmeras de segurança, o TZS possibilita uma estimação de movimento eficaz sem sobrecarregar o sistema. Embora possa ocorrer uma ligeira perda de precisão em comparação ao FS, o TZS apresenta um equilíbrio muito bom entre precisão, velocidade e eficiência de processamento, representando, portanto, uma escolha superior para a maioria das aplicações práticas (Goncalves et al., 2023).

O TZS é estruturado em várias etapas, cada uma com um papel específico no processo de busca e otimização do movimento em vídeos. As etapas executadas são as seguintes:

1. **Predição de Vetor de Movimento** (Seção 2.4.0.1): Esta etapa envolve a predição do vetor de movimento inicial de um bloco atual, utilizando os vetores de movimento de blocos adjacentes. É uma fase importante para estabelecer um ponto de partida para as buscas subsequentes;

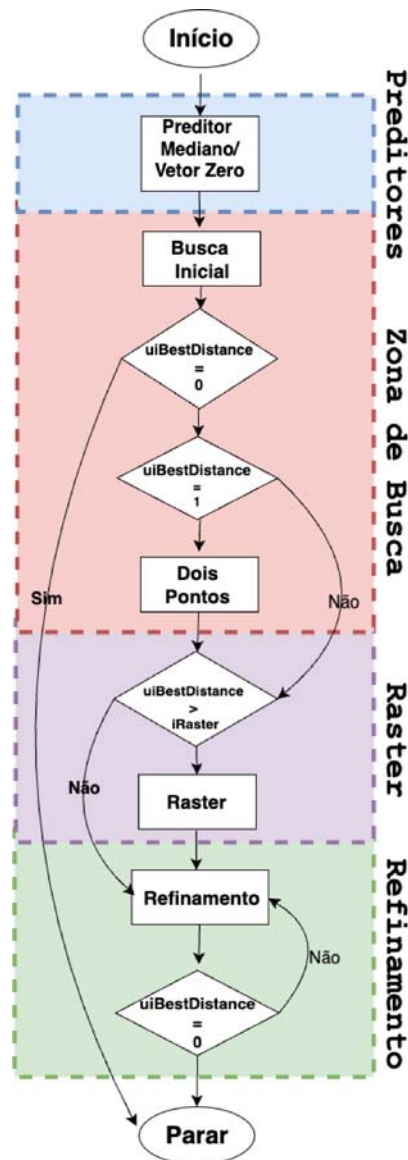
2. **Busca Inicial** (Seção 2.4.1): Após estabelecer um ponto de partida, realiza-se uma busca preliminar em uma área predefinida, pela etapa anterior, usando um padrão em diamante ou quadrático. Esta fase é fundamental para o sucesso do algoritmo, pois cria as bases para as etapas seguintes, concentrando-se em áreas com potencial para a correspondência de movimento ideal. Identificando regiões promissoras, o algoritmo direciona suas etapas subsequentes de busca para essas áreas, otimizando a eficiência e a precisão na estimativa de movimento.
3. **Busca de Dois Pontos** (Seção 2.4.2): Nesta fase, são avaliados dois pontos adicionais localizados de forma adjacente ao melhor vetor de movimento previamente identificado, pela etapa anterior (Gogoi; Peesapati, 2020), no caso *Diamond Search*.
4. **Busca em Varredura** (Seção 2.4.3): A busca em varredura (*Varredura*) é ativada quando a distância entre o melhor candidato e a posição inicial é maior Varredura. A busca Varredura, uma versão reduzida da *full search*.
5. **Busca Refinada** (Seção 2.4.4): É a etapa final do TZS, que busca aprimorar o vetor de movimento obtido nas etapas anteriores. Uma nova busca é realizada em torno do melhor ponto identificado, novamente seguindo um padrão de busca em diamante ou quadrático.

#### 2.4.0.1 Preditor de Movimento no TZS

O algoritmo TZS de busca de movimento utiliza componentes chave para estimar o vetor de movimento de um bloco atual, destacando-se o preditor mediano e o vetor zero. A finalidade principal desses componentes é aproveitar os vetores de movimento de blocos adjacentes, bem como o vetor zero, para fornecer estimativas precisas do movimento entre quadros. O processo inicia-se com a seleção dos blocos adjacentes ao bloco em foco, que inclui os blocos localizados à esquerda, acima e na diagonal superior direita, além do próprio vetor zero. Em seguida, os vetores de movimento desses blocos são adquiridos, servindo como base para a estimativa do movimento. O cálculo do preditor mediano envolve a determinação da mediana dos quatro vetores de movimento, processada separadamente para as componentes horizontal e vertical. A título de exemplo, se os vetores são  $A = (x_1, y_1)$ ,  $D = (x_3, y_3)$  e  $L = (x_2, y_2)$ , a mediana horizontal é calculada entre  $x_1$ ,  $x_2$ , e  $x_3$ , e de forma análoga para a componente vertical com  $y_1$ ,  $y_2$ , e  $y_3$  e o vetor zero.

A fase de predição de vetor de movimento é essencial no TZS, constituindo a base para as etapas subsequentes de busca de movimento. O TZS aplica os quatro preditores básicos oriundos dos blocos adjacentes ao bloco em análise. Esses preditores

Figura 10 – Etapas do Test Zone Search. Adaptado de (Doan et al., 2017)



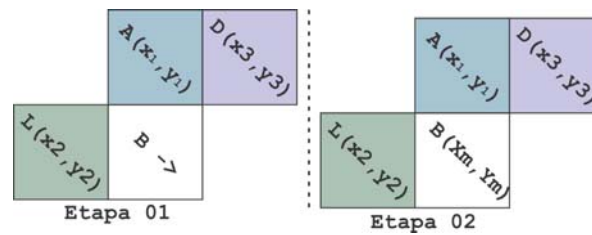
são categorizados conforme sua localização espacial: adjacente à esquerda, adjacente acima e na diagonal superior direita (Goncalves et al., 2023). A escolha desses preditores antecede o cálculo do preditor mediano.

A Figura 11 exemplifica esse processo, demonstrando o cálculo do vetor de movimento do bloco B. Os vetores de movimento dos blocos A, L e D são representados por  $(x1, y1)$ ,  $(x2, y2)$  e  $(x3, y3)$ , na etapa 1, respectivamente. As componentes dos vetores são organizadas e a mediana de cada uma é determinada, resultando no vetor de movimento final para o bloco  $B(X_m, Y_m)$ , na etapa 2.

Após a determinação do preditor mediano, o algoritmo prossegue para a avaliação com o vetor zero. Neste estágio, a eficácia de cada preditor é mensurada utilizando-se o SAD. A escolha do preditor mais eficiente se dá pelo critério do menor valor de SAD, garantindo que este seja utilizado nas fases subsequentes do processo de busca de movimento, especificamente no TZS.



Figura 11 – Preditores do TZS



### 2.4.1 Busca Inicial

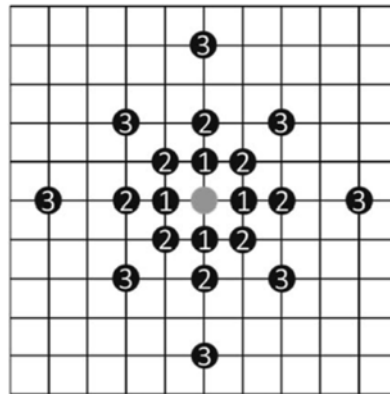
A segunda etapa do TZS, chamada Busca Inicial, é crucial para o processo. Aqui, o objetivo principal é explorar a área em torno do ponto de partida determinado pela etapa anterior, a Predição de Vetor de Movimento. Essa predição se baseia em estudos de Bross (2021), Pakdaman (2020), e outros (Bross et al., 2021; Pakdaman; Hashemi; Ghanbari, 2020; Loukil; Mayet, 2023). A eficácia dessa etapa é vital, pois ela define a região de busca para as próximas etapas, o que influencia diretamente na eficiência do algoritmo.

O processo inicia com a definição de um ponto de partida, escolhido com base no melhor preditor de vetor de movimento previamente selecionado. Em torno desse ponto, estabelece-se a área de busca, onde são aplicados padrões específicos de busca, como os formatos de diamante e quadrado. Dentro dessa área, cada ponto dos padrões mencionados é avaliado como um potencial candidato para o vetor de movimento ideal. Utiliza-se uma métrica de distorção, como a Soma das Diferenças Absolutas (SAD), com o objetivo de identificar o ponto que apresenta o menor valor de distorção. Este ponto é então escolhido como o melhor, e a distância deste em relação ao ponto de partida é registrada na variável *uiBestDistance*. Esta variável é essencial para as etapas subsequentes do algoritmo, influenciando diretamente a decisão de prosseguir para a Busca de Varredura.

Após concluir a Busca Inicial, que pode envolver entre 1 a 64 expansões do padrão de busca dependendo dos resultados, encerra-se essa fase. Se nenhum novo ponto se mostrar superior ao candidato anterior, a busca é finalizada. Uma representação visual deste processo pode ser vista na Figura 12, que exemplifica a Busca Inicial do algoritmo TZS no software VVC (VTM) (HEVC TEST MODEL, HM; , jvet; Feldmann, 2020). Na primeira expansão, o padrão de diamante possui 4 pontos, e nas expansões subsequentes, expande-se para 8 pontos, totalizando 20 pontos avaliados.

O valor da *uiBestDistance* determina os próximos passos do algoritmo: se for igual a um, a busca segue para a etapa de dois pontos; caso contrário, procede-se para a etapa de busca Varredura ou *raster*.

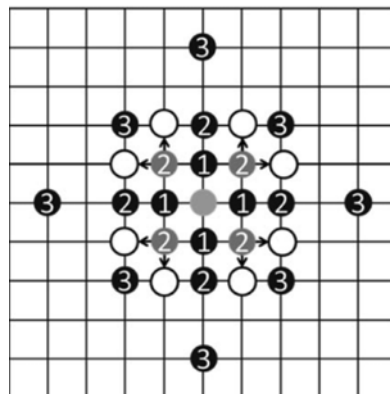
Figura 12 – Busca Inicial do TZS. (Doan et al., 2017)



### 2.4.2 Busca de Dois Pontos

A Busca de Dois Pontos constitui a terceira etapa no algoritmo TZS, conforme detalhado por Doan (2017) (Doan et al., 2017) e (Gogoi; Peesapati, 2020). Esta fase é iniciada quando se verifica a necessidade de uma busca mais refinada, geralmente após a constatação de que a variável *bestdistance*, determinada na etapa anterior, é igual a um. Esse critério é aplicado após a conclusão da Busca Inicial, indicando que a posição de melhor correspondência necessita de uma investigação mais detalhada para otimizar o alinhamento do movimento. A ilustração na Figura 13 exemplifica todos os casos possíveis de uma busca de dois pontos. Essa visualização ocorre imediatamente após a primeira busca de diamante. Caso a melhor correspondência seja identificada em um dos quatro pontos de busca cinza com um comprimento de passo de 2, dois pontos adicionais de busca são gerados ao redor dessa correspondência, marcados por círculos em branco. Essa etapa adicional ajuda a refinar ainda mais a precisão do vetor de movimento, garantindo a seleção da melhor correspondência possível.

Figura 13 – Busca Dois Pontos . (Doan et al., 2017)

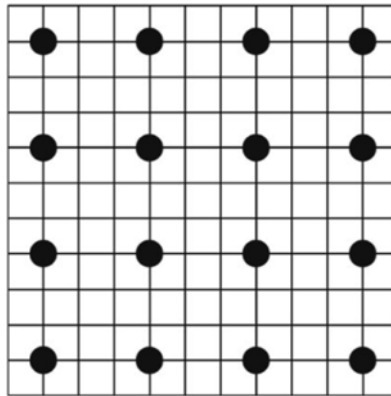


### 2.4.3 A Etapa de Busca de Varredura (*Raster*)

A busca *Varredura* é um método eficiente que se inicia quando a distância da melhor correspondência, representada pela variável *uiBestDistance*, excede o valor do parâmetro *iraster*, o valor padrão de *iraster* é 5. Esse critério é importante para otimizar o processo de busca, particularmente em situações em que a correspondência inicial encontra-se consideravelmente afastada do centro de busca atual. A busca *Varredura* utiliza uma estratégia de subamostragem, diferentemente da busca completa, conhecida como *full search*. Isso implica na diminuição do número de passos para o valor de *Varredura*, uma medida que acelera o processo de busca sem comprometer significativamente a precisão desejada. Esta técnica é vantajosa por economizar recursos computacionais e tempo. Conforme a busca *Varredura* progride, espera-se que o valor de *uiBestDistance* se ajuste ao de *iraster*, sinalizando a conclusão da busca dentro do limite de passos estabelecido. A eficácia deste método pode ser visualizada na Figura 14, que exemplifica o processo quando *iraster* é ajustado para 3. A ilustração destaca a redução no número de passos e a efetiva cobertura da área de busca dentro do intervalo definido, oferecendo uma compreensão visual clara do procedimento. Esta representação gráfica não apenas esclarece a execução da busca *Varredura*, mas também demonstra sua eficiência em contextos práticos.

$$npontos = \left\lceil \frac{(tBloco - tBusca + 1) \cdot 2}{i_{Varredura}} \right\rceil^2 \quad (4)$$

Figura 14 – Busca Varredura. (Doan et al., 2017)



### 2.4.4 Busca Refinada

A etapa final do TZS é a Busca Refinada (Ref), que tem como objetivo aprimorar o vetor de movimento obtido nas etapas anteriores, conforme discutido por (Bross et al., 2021) e (Pakdaman; Hashemi; Ghanbari, 2020). Essa etapa é importante para aumentar a qualidade da estimativa de movimento, impactando positivamente na eficiência de codificação.

O ponto de partida para Busca Refinada (Ref) é o melhor ponto identificado na etapa anterior, seja na Busca Inicial ou na Busca de Varredura. Uma nova busca é realizada em torno desse ponto, geralmente seguindo um padrão de busca similar ao utilizado na primeira etapa do algoritmo, ou seja, em forma de diamante ou quadrática. Esse padrão pode ser expandido em dois níveis para explorar os pontos vizinhos em busca de uma correspondência ainda mais precisa.

Uma métrica de distorção, como o SAD, é novamente utilizada para avaliar os pontos candidatos. O objetivo é encontrar um ponto com um valor de distorção menor do que o melhor ponto identificado anteriormente.

Assim, Busca Refinada (Ref) atua como uma etapa de otimização final no TZS, sendo fundamental para a precisão da estimativa de movimento e, conseqüentemente, para a eficácia geral do algoritmo.

### 3 COMPUTAÇÃO APROXIMADA

Esse capítulo irá, explorar o conceito de computação aproximada e sua aplicação na codificação de vídeo, focando em operadores aritméticos como somadores e subtratores. Ele aborda a relevância e o impacto da aplicação desses operadores na eficiência energética e no desempenho de sistemas de codificação de vídeo. Além disso, discute a oportunidade de pesquisa existente na área de subtratores aproximados e a sua importância em contextos onde as subtrações são frequentes, como no cálculo da métrica de distorção em codificadores de vídeo. O capítulo destaca a necessidade de um equilíbrio entre a precisão computacional e a eficiência energética, especialmente em aplicações multimídia, que toleram certas imprecisões. A computação aproximada (*approximate computing*), é um campo emergente na pesquisa tecnológica, e responde à crescente demanda por eficiência energética em sistemas computacionais. Segundo Petroc (Taylor, 2023), em 2021, o número de usuários de dispositivos móveis no mundo estava em 7,1 bilhões. Em 2025, espera-se que o número de usuários de dispositivos móveis no mundo atinja 7,49 bilhões. Sendo que, em média, um celular é carregado a 0,15 quilowatts-hora de eletricidade por mês e 1,83 quilowatts-hora de eletricidade por ano (Marsh, 2023). Assim, em 2025 teremos um consumo aproximado de 13,71 bilhões de quilowatts-hora (Marsh, 2023). Isso equivale a 19,6% da produção anual da Itaipu Binacional no ano de 2022 (Itaipu, 2024). A computação aproximada vem como um recurso que pode ser usado para reduzir o consumo energético em aparelhos móveis.

Essa abordagem oferece um equilíbrio entre precisão e eficiência energética. Ao aceitar resultados aproximados em vez de exatos, benefícios podem ser alcançados, incluindo redução de área do hardware, economia de energia e aumento da velocidade de processamento (Davidson, 2016; Kahng; Kang, 2012).

Contudo, a aplicação da computação aproximada envolve desafios importantes. É crucial identificar as aplicações adequadas para essa técnica e definir o intervalo de tolerância para os resultados aproximados. Algumas aplicações não suportam resultados não exatos, e mesmo aquelas tolerantes a erros podem ter componentes críticos onde imprecisões podem resultar em falhas graves. Assim, uma etapa fun-

damental na implementação dessa técnica é discernir quais componentes do sistema são tolerantes a erros e estabelecer o grau de aproximação aceitável para a aplicação em questão (Xu; Mytkowicz; Kim, 2016). Venkataramani (Venkataramani et al., 2015) ressalta aspectos importantes a considerar na implementação da computação aproximada, incluindo:

- **Aceitabilidade dos Resultados:** Na computação aproximada, a definição clara de resultados aceitáveis para cada aplicação é essencial. É necessário desenvolver estratégias que garantam a manutenção da qualidade da saída, mesmo com a implementação de cálculos aproximados;
- **Resiliência a Erros:** Sistemas que empregam computação aproximada devem ser capazes de operar corretamente apesar de resultados inexatos ou aproximados, evidenciando a necessidade de resiliência a erros;
- **Intervalo de Valores Aproximados:** É fundamental estabelecer um intervalo de tolerância para valores aproximados, variando conforme o tipo e a complexidade da aplicação.

A computação aproximada encontra aplicações em áreas como *Big Data*, computação gráfica, processamento de sinais, visão computacional, *Deep Learning* e segurança biométrica (Shafique et al., 2016). Essas áreas compartilham características como redundância de dados, tolerância a resultados aproximados e, frequentemente, a interação com seres humanos como usuários finais.

Especificamente, a computação multimídia se destaca no uso da computação aproximada (Gupta et al., 2013; Raha; Jayakumar; Raghunathan, 2014; Gupta et al., 2011; Porto et al., 2020). Devido à percepção humana limitada, aplicações multimídia muitas vezes não exigem precisão numérica absoluta, como em algoritmos de processamento de imagem e vídeo.

Reda (Reda; Shafique, 2018) destaca áreas onde a computação aproximada tem sido eficaz:

- **Big Data:** A natureza inerentemente ruidosa e aproximada dos dados em grande escala torna o *Big Data* um campo propício para a aplicação dessa técnica;
- **Computação Gráfica:** A computação aproximada pode otimizar processos como renderização de alta qualidade, que são intensivos em recursos;
- **Processamento de Sinais:** A gestão de erros inerentes às entradas sensoriais discretas no processamento de sinais é uma aplicação potencial da computação aproximada;

- **Visão Computacional:** Algoritmos de visão computacional, que naturalmente possuem limitações de precisão, podem se beneficiar da eficiência proporcionada pela computação aproximada;
- **Deep Learning:** A tolerância a erros das redes neurais em *Deep Learning* torna essa área ideal para a aplicação da computação aproximada;
- **Segurança Biométrica:** A resiliência a variações nas entradas, como na identificação biométrica, é um exemplo de como a computação aproximada pode melhorar a eficiência do sistema.

A validação de resultados aproximados é essencial na computação aproximada. Comparar os resultados aproximados com os exatos ajuda a estabelecer um intervalo de tolerância aceitável. Algumas métricas comuns para avaliar a computação aproximada incluem:

- **Distância Absoluta (DA):** Mede a diferença absoluta entre o resultado correto ( $R_c$ ) e o aproximado ( $R_i$ ), conforme a equação 5.

$$DA = |R_c - R_i| \quad (5)$$

- **Precisão (Pre):** Avalia a precisão da operação, com valores entre 0% e 100%, conforme a equação 6.

$$Pre = (1 - (DA/R_c)) \times 100\% \quad (6)$$

- **Precisão Mínima Aceitável (PMA.):** Representa o valor de precisão mínimo aceitável para o resultado aproximado.
- **Probabilidade Aceita (PA):** Mede a proporção de resultados aproximados dentro do intervalo aceitável, expressa por  $PA = P(Pre > PMA.)$ .

### 3.1 Níveis de Abstração

A computação aproximada é um conceito abrangente que se aplica em vários níveis de abstração, incluindo hardware e software. Esta técnica é fundamental para otimizar o uso de recursos, reduzindo a precisão ou a fidelidade em partes do sistema onde a exatidão absoluta não é crítica. Este capítulo explora as diversas facetas da computação aproximada, desde a sua aplicação em circuitos e arquiteturas até o nível de software e algoritmos.

- **Hardware Aproximado**

No âmbito do hardware, a computação aproximada se manifesta através de técnicas como *pruning* lógico e escalonamento de tensão em CMOS probabilístico.

O CMOS (Complementary Metal-Oxide-Semiconductor) probabilístico é uma técnica emergente no campo de hardware aproximado que visa otimizar o consumo de energia e a eficiência dos circuitos eletrônicos, especialmente em dispositivos portáteis e sistemas embarcados. Diferente do CMOS tradicional, que opera com precisão determinística, o CMOS probabilístico introduz uma abordagem que permite erros controlados nas operações de computação para alcançar ganhos significativos em termos de consumo de energia e desempenho, baseadas no princípio de design orientado por significância (Liu; Lombardi; Shulte, 2020). Estas técnicas visam reduzir o consumo de energia, alocando recursos de forma diferenciada com base na importância dos bits. Por exemplo, em circuitos aritméticos aproximados, menos recursos são alocados para partes de menor significância. Outras abordagens incluem o uso de somadores especulativos e multiplicadores aproximados (Seiler; Taherinejad, 2023), onde a precisão é balanceada com o desempenho e o consumo de energia.

A memória aproximada representa outra vertente significativa, onde diferentes técnicas são aplicadas para reduzir o consumo de energia, como alterar a taxa de atualização para DRAM (Liu et al., 2023) e modificar a tensão de fornecimento em SRAM. Essas abordagens priorizam a eficiência energética, sacrificando a precisão em dados menos críticos.

- **Software e Algoritmos Aproximados**

No nível de software e algoritmos, a computação aproximada frequentemente envolve a escala de precisão. Isso implica em reduzir a largura de bits de operandos para economizar recursos computacionais e de armazenamento. Técnicas em *loops* também são exploradas (Dou et al., 2023), para reduzir o tempo de computação e economizar energia. Em um nível mais alto, programas podem optar por omitir certas tarefas e acessos à memória para otimizar a performance em arquiteturas multicore (Fatemieh; Reshadinezhad; Taherinejad, 2023).

- **Ferramentas para Computação Aproximada**

Diversas ferramentas foram desenvolvidas para avaliar e aplicar a computação aproximada. No projeto de hardware, as técnicas de aproximação são adaptadas com base nos parâmetros do projeto. No software, a automação é utilizada para otimizar motores gráficos e outras aplicações, com *kernels* aproximados (Wen et al., 2023). Compiladores e linguagens de programação que suportam computação aproximada permitem especificar dados aproximados e aplicar técnicas como armazenamento em memória aproximada e processamento por unidades aritméticas inexatas (Mishra et al., 2023).

A computação aproximada é um campo multidisciplinar que engloba diversas téc-



nicas e abordagens, cada uma com seu próprio conjunto de compromissos. Desde o projeto de circuitos e arquiteturas até o desenvolvimento de software, essa abordagem oferece um caminho promissor para melhorar a eficiência energética e o desempenho de sistemas computacionais, com aplicações em diversas áreas da tecnologia.

## 3.2 Operadores Aritméticos Aproximados

A aplicação de computação aproximada em operadores aritméticos tem grande potencial de trazer resultados importantes. Em circuitos digitais, operações básicas, como soma, subtração, multiplicação e divisão, são essenciais. Os operadores aproximados combinam elementos precisos e aproximados, geralmente aplicando precisão aos bits mais significativos e aproximação aos menos significativos. O projeto desses operadores envolve a simplificação da arquitetura, seja em nível de portas lógicas, de transistores ou modificando a Tabela verdade de um operador funcional (Han; Orshansky, 2013).

### 3.2.1 Somadores Aproximados

Os somadores são operadores essenciais, pois não necessitam de outros operadores para a sua criação, ao contrário de multiplicadores, divisores e operadores de raiz quadrada, por exemplo. Eles são amplamente utilizados em circuitos de processamento de sinais. Vários pesquisadores (Jiang; Han; Lombardi, 2015; Jiang et al., 2017; Mahdiani et al., 2010; Shafique et al., 2015) classificam os somadores aproximados de acordo com a metodologia empregada para implementar a aproximação. As categorias comumente adotadas incluem somadores especulativos, somadores segmentados, somadores *Carry-Select* e somadores completos aproximados.

1. **Somadores Especulativos:** Esses somadores utilizam a previsão do *carry* baseada nos bits menos significativos. Um exemplo são os somadores descritos em (Verma; Brisk; lenne, 2008), que operam sob a premissa de que a sequência mais longa de propagação de sinais tem, em média, um comprimento de  $\log n$ , com  $n$  representando a largura em bits dos números inteiros somados. Esses somadores são eficazes, apresentando resultados corretos em 99,99% dos casos, e são compostos por uma estrutura chamada somador especulativo de latência variável (*Variable Latency Speculative Adder* - VLSA), que identifica a propagação do sinal de *carry*, e o somador quase correto (*Almost Correct Adder* - ACA), que realiza a soma;
2. **Somadores Segmentados:** Esses somadores dividem a sequência de bits em blocos, realizando a adição de forma independente em cada bloco. Os resultados são, então, combinados, com prioridade para os bits mais significativos (MSBs).

Essa abordagem permite uma operação mais rápida, embora possa introduzir erros nos bits menos significativos;

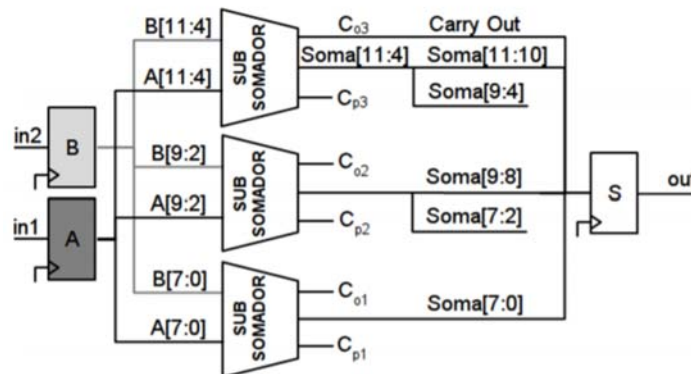
3. **Somadores *Carry-Select*:** São compostos por vários subsomadores, cada um responsável por somar uma sequência de  $m$ -bits em paralelo. Essa estrutura permite uma operação rápida, mas pode aumentar o consumo de energia e a área do circuito;
4. **Somadores Completos:** Dividem as entradas de  $n$ -bits em duas partes, garantindo a precisão do bit mais significativo (MSB), enquanto os bits menos significativos (LSBs) são aproximados. Essa abordagem é útil em aplicações onde a precisão dos LSBs é menos crítica.

Os somadores aproximados têm sido muito explorados na literatura e há uma vasta quantidade de publicações sobre eles. Nesta seção, apresentamos alguns somadores aproximados e suas estratégias de utilização da aproximação:

- **Somador Configurável de Precisão Genérica** (Figura 15): O somador configurável de precisão genérica (*Generic Accuracy Configurable Adder* - GeAr) (Shafique et al., 2015), destaca-se pela sua configurabilidade na definição da precisão. Esse dispositivo utiliza unidades de subsomadores de  $K$  bits que operam em paralelo ( $L$ ), com o comprimento dos subsomadores sempre menor ou igual a  $N$  (i.e.,  $L \leq N$ ), objetivando a redução do comprimento máximo da propagação do transporte de *carry* (Chippa et al., 2013). Os somadores são compostos por dois tipos de bits: (1) bits  $R$ , resultantes, que contribuem para a soma final; e (2) bits  $P$ , de predição, usados para prever o transporte do *carry* para a resultante de bits. Cada somador contribui com um total de bits  $R$  para a saída, com exceção do primeiro somador, que gera  $L = R + P$  número de bits para criar a saída para os bits  $P$ , que são os menos significativos.
- **Power-Precision Scalable Adder** (Figura 17): O Operador 2PSA é uma arquitetura de somador aritmético configurável que permite ajustar a precisão e o consumo energético conforme a necessidade da aplicação. Esta arquitetura suporta modos de operação precisos e imprecisos, com os imprecisos introduzindo erros controlados para reduzir o consumo de energia. Utiliza dois subsomadores com isolamento de operandos para otimizar a dissipação energética e o desempenho, sem sobrecarregar o hardware.

A Figura 16 exibe o diagrama em blocos do 2PSA de 4 bits, mostrando os componentes chave: os blocos OI  $n$  para isolamento de operandos, IAn como operadores imprecisos de 1 bit, e PA  $n$  como operadores precisos de 1 bit. A entrada PQControl ajusta a combinação de saídas de IAn e PA  $n$  para diferentes níveis

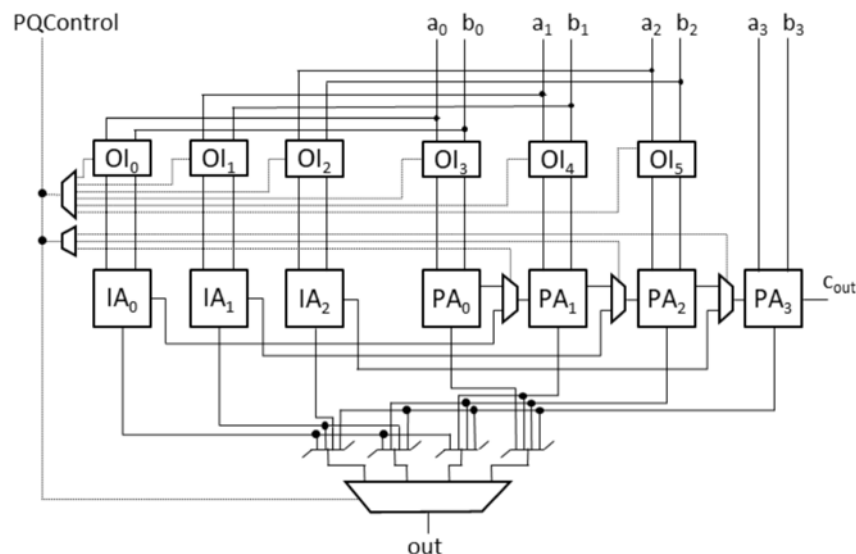
Figura 15 – Somador Aproximado GeAr (Shafique et al., 2015). Adaptado: (Roger Endrigo Carvalho Porto, 2020)



de precisão e controla o carry in no PA  $n$ , além de ser crucial para o isolamento de operandos através dos blocos  $OI_n$ .

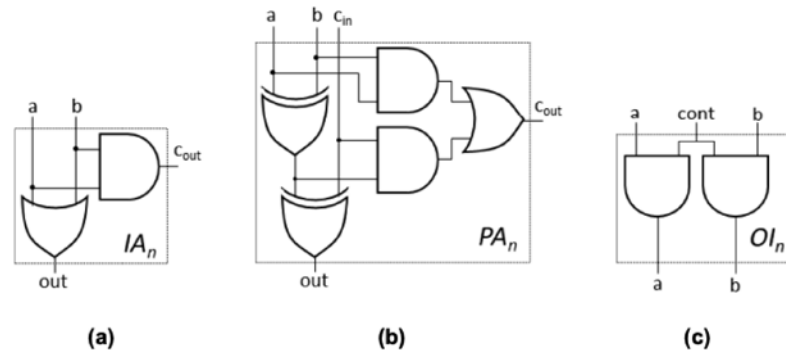
Como demonstrado na Figura 17, a implementação do 2PSA inclui componentes adicionais que possibilitam a seleção entre diferentes pontos de operação. A configuração para operações precisas pode aumentar a dissipação de potência, mas os modos imprecisos permitem uma redução significativa no consumo energético em comparação com operadores simples como o RCA ou o LOA.

Figura 16 – Somador Aproximado Completo 2PSA (Porto et al., 2020)



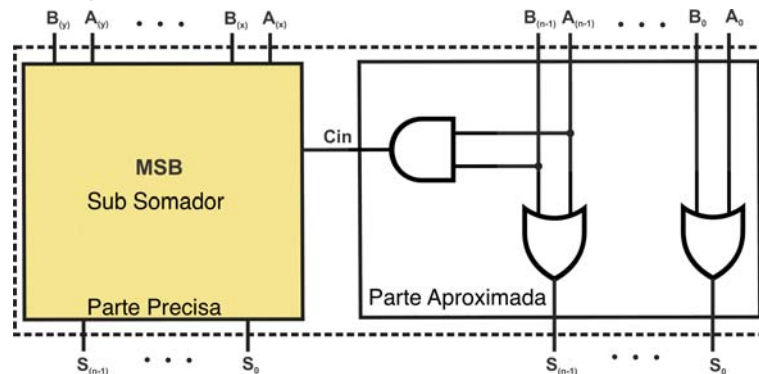
- **Somador OR na Parte Inferior** (Figura 18): O somador OR na parte inferior (*Lower-part-OR* - LOA) (Mahdiani et al., 2010) adota uma estratégia distinta, dividindo uma adição de  $k$  bits em duas partes menores, ou seja, em dois módulos de  $m$ -bits e  $n$ -bits. O módulo de  $m$ -bits do LOA emprega somadores precisos (subsomadores) para calcular os  $m$ -bits mais significativos do resultado,

Figura 17 – Somador Aproximado 2PSA (Porto et al., 2020)



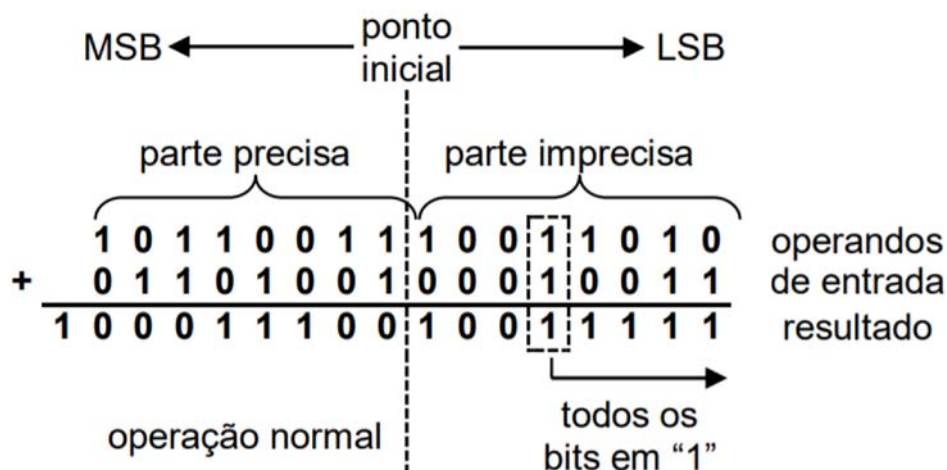
enquanto na parte imprecisa ( $n$ -bits menos significativos) são executadas operações de *OR* bit a bit, ao invés da operação *XOR*, introduzindo imprecisão. Uma porta *AND* adicional é utilizada para gerar a carga para o subsomador preciso quando os bits mais significativos de ambas as entradas na parte inferior são um. A não consideração dos sinais de transporte na parte inferior do somador *LOA* pode acarretar em perda elevada de precisão.

Figura 18 – Somador Aproximado LOA (Mahdiani et al., 2010).



- **Somador Tolerante a Erros - I** (Figura 19): O somador tolerante a erros - I (*Error Tolerant Adder- ETA I*) (Zhu et al., 2010) emprega uma abordagem híbrida, combinando blocos aproximados e precisos. No bloco aproximado, é eliminada a propagação de *carry*, utilizando-se um meio somador e uma porta lógica *AND* para cada bit do somador. A soma dos bits é realizada dos mais significativos para os menos significativos. Cada bit é processado por somadores até que o valor um seja observado simultaneamente em ambas as entradas do somador, resultando em uma resposta um. Após esse ponto, todos os bits subsequentes são automaticamente definidos como um. Essa técnica do ETA-I possibilita uma aproximação tolerante a erros, sendo vantajosa em aplicações onde a precisão absoluta não é essencial.
- **Somador Tolerante a Erros - II** (Figura 20): O somador ETA-II foi desenvolvido para aprimorar o desempenho do ETA I (Zhu et al., 2010), o ETA II aborda a

Figura 19 – Somador Aproximado ETA-I (Mahdiani et al., 2010). Adaptado: (Roger Endrigo Carvalho Porto, 2020)



limitação do ETA I em cálculos com entradas de valores menores. Por exemplo, ao somar números de 16 bits como 001111 (15) + 001111 (15), o ETA I resulta em 001111 (15) ao invés do correto 011110 (30). O ETA II busca melhorar essa aproximação. A soma de  $N$ -bits é dividida em  $M$  ( $M \geq 2$ ) blocos, com cada bloco contendo  $N/M$  bits. Cada bloco possui dois circuitos distintos: um gerador de *carry* e um gerador de soma. Um bloco gera o *carry* de saída para o bloco subsequente, enquanto o outro gera a soma. Apesar de utilizar circuitos precisos em sua implementação em hardware, o ETA II é classificado como um somador aproximado, pois todos os blocos geradores de *carry* assumem o *carry* de entrada como zero. O bloco gerador de soma é um somador completo do tipo *Ripple Carry Adder* (RCA). Já o bloco gerador de *carry* implementa um somador tipo *Carry Look-ahead Adder* (CLA).

- **Somador Cópia** (Figura 21): A técnica de aproximação por cópia, proposta por Gupta (Gupta et al., 2013) em seu somador Copy, emprega o método *bypass*. Nesse método, uma das entradas, A ou B, é transferida diretamente para a saída. Esse procedimento resulta em um resultado correto em 50% das oito possíveis combinações para operações de 1-bit. O somador é segmentado em duas partes: os bits mais significativos, onde são utilizados somadores exatos de uso geral, como RCA, e os bits menos significativos, constituindo a parte imprecisa.
- **Somador Aproximado com Precisão Configurável** (Figura 22): O somador aproximado com precisão configurável (*Accuracy Configurable Approximate Ad-*

Figura 20 – Somador Aproximado ETA II (Mahdiani et al., 2010). Adaptado: (Roger Endrigo Carvalho Porto, 2020)

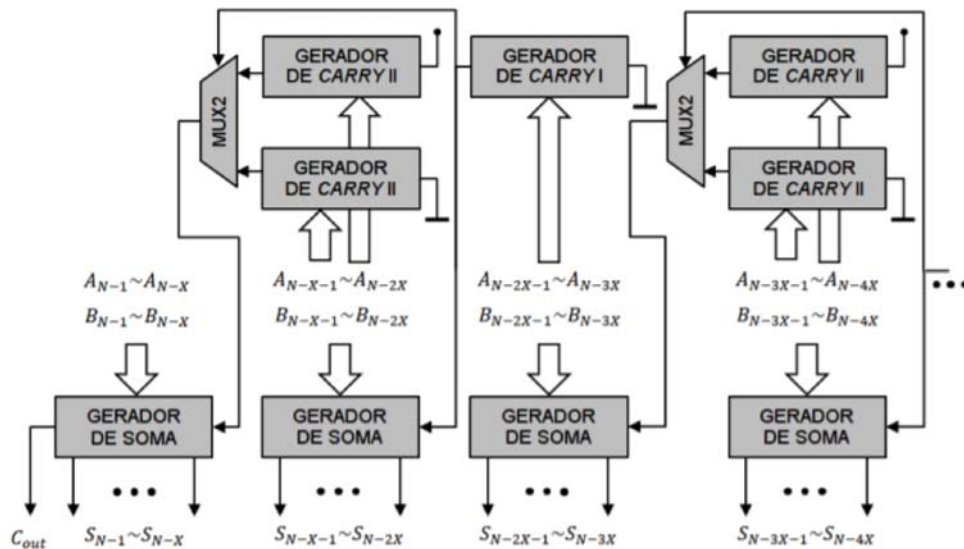
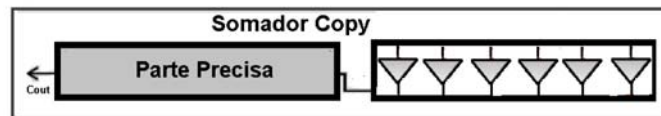


Figura 21 – Somador Aproximado Copy. Adaptado por (Macedo et al., 2018).



der - ACAA) (Kahng; Kang, 2012) processa suas entradas dividindo-as em sub-blocos sobrepostos. Cada bloco é responsável por calcular a soma dos bits mais significativos, cujos resultados são concentrados na saída final. Essa abordagem torna o operador mais rápido, devido à menor propagação de *carry* e à realização da soma de forma paralela. Os blocos responsáveis pelos bits mais e menos significativos utilizam somadores precisos, enquanto para os bits intermediários são empregados blocos de somadores aproximados.

- **Somador Completo de 1-bit** (Figura 23): O somador completo de 1-bit (*1-bit Full Adder* - 1-bit FA) foi desenvolvido como uma otimização no RCA (Kedem et al., 2011), e é uma inovação importante. O RCA é composto por uma sequência de vários somadores completos, cada um formado por duas portas *XOR*, duas portas *AND* e uma *OR*. Essa estrutura foi simplificada para incluir apenas duas portas *XOR* e um multiplexador. A Tabela verdade para essa configuração indica que, de oito operações possíveis, quatro resultam em imprecisões.

As diferentes metodologias de somadores aproximados, cada uma com suas características únicas, destacam a diversidade e a adaptabilidade desses operadores. A habilidade de configurar precisão e eficiência energética em um equilíbrio otimizado

Figura 22 – Somador Aproximado ACAA (Kahng; Kang, 2012). Adaptado: (Roger Endrigo Carvalho Porto, 2020)

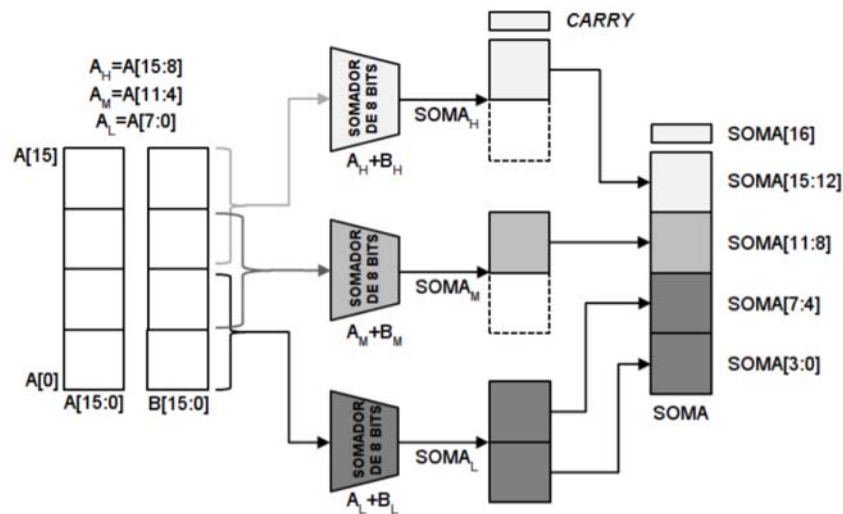
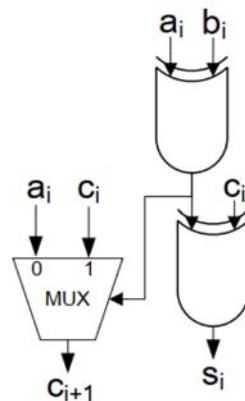


Figura 23 – Somador Aproximado 1-bit FA (Kedem et al., 2011).



é o que torna os somadores aproximados elementos-chave na inovação tecnológica, contribuindo significativamente para o desenvolvimento de tecnologias emergentes e para a sustentabilidade em sistemas eletrônico

### 3.2.2 Subtratores Aproximados

Os subtratores representam uma classe essencial de operadores em uma vasta gama de sistemas digitais. Embora sejam menos prevalentes em comparação aos somadores, seu papel é incontestável em certas aplicações. Frequentemente, subtratores são substituídos por somadores que empregam o método do complemento de dois. Essa técnica consiste em transformar uma das entradas em valor negativo, sendo uma abordagem largamente adotada na aritmética de computadores para realizar subtrações (Koren, 2003).

No entanto, a utilização de subtratores torna-se indispensável em sistemas digitais especializados em operações extensivas de subtração. A opção por um somador



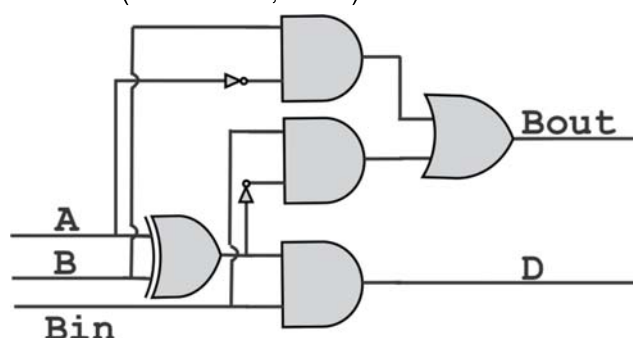
aproximado que utiliza o complemento de dois, apesar de parecer benéfica inicialmente, resulta em perdas significativas de recursos. Essas perdas se manifestam de diversas formas, como aumento da área ocupada pelo circuito, redução da velocidade de processamento e elevação do consumo de energia. Contrastando com isso, um subtrator aproximado demanda menos recursos, uma vez que o somador aproximado necessita de um número adicional de portas lógicas para efetuar uma subtração.

Dado que o foco deste trabalho está exatamente nos subtratores aproximados, alguns dos operadores publicados na literatura serão discutidos em mais detalhes nesta seção do texto.

- **Subtrator com Uma Célula Subtratora Exata (Figuras 24, 25 e 26):** Chen et al. (Chen et al., 2015) desenvolveram divisores aproximados para aplicações que toleram erros, incluindo subtratores aproximados. Os autores propuseram projetos que atenuam a complexidade lógica, reduzem o número de transistores, o consumo de energia e o atraso dos subtratores, aceitando um grau de imprecisão. Foram três versões do subtrator batizado como subtrator com uma célula subtratora exata (*An Exact Subtractor Cell - AXSC*): AXSC1, AXSC2 e AXSC3. Empregando tecnologia CMOS de 45 nm, os subtratores foram avaliados com base em métricas de erro. A eficácia desses subtratores foi demonstrada em aplicações onde a exatidão absoluta não é primordial.

Cada versão dos subtratores AXSC propostos por Chen et al. (Chen et al., 2015) é caracterizada por uma configuração única de portas lógicas: sete para a versão AXSC 1, apresentado na Fig. 24, duas para a versão AXSC 2, apresentado na Figura 25 e seis para a versão AXSC 3, apresentado na Figura 26. O desempenho desses subtratores é quantificado pela distância absoluta do erro, sendo dois para as versões 1 e 3, e quatro para a versão 2. Esses aspectos serão explorados com mais detalhes nas seções seguintes do trabalho.

Figura 24 – Subtrator AXSC 1 (Chen et al., 2015).



- **Subtrator aproximado:** os subtratores aproximados (*Subtractor Imprecise - ApXs*) foram projetados para serem incorporados em uma arquitetura rápida de transformada de Fourier. Esses subtratores, apresentados na Fig. 27 e na Fig.



Figura 25 – Subtrator AXSC 2 (Chen et al., 2015).

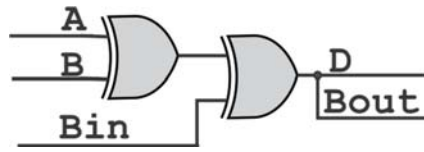
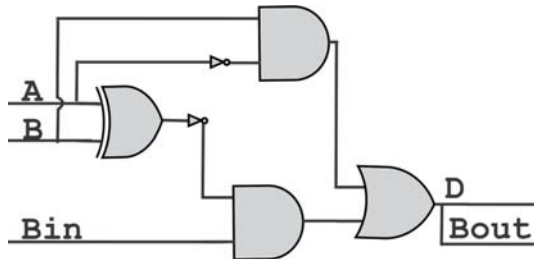
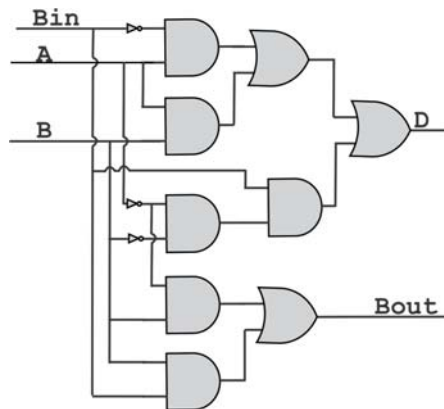


Figura 26 – Subtrator AXSC 3 (Chen et al., 2015).



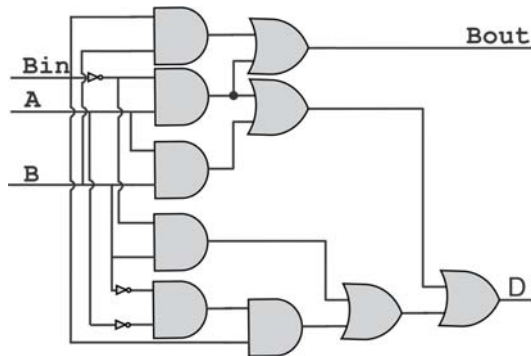
28, foram desenvolvidos em duas versões e são compostos por um conjunto de portas lógicas *OR*, *AND* e um inversor. Esses subtratores visam otimizar o desempenho e reduzir a complexidade em aplicações de processamento de imagens. Uma técnica de minimização lógica foi empregada pelos autores para projetar e validar um somador *Ripple Carry* de 8 bits e um Subtrator *Ripple Borrow*. Este estudo também destaca o potencial dos memristores na computação aproximada, oferecendo uma solução promissora para a implementação eficiente em dispositivos nanoeletrônicos.

Figura 27 – Subtrator ApXs (Muthulakshmi; Dash; Prabakaran, 2018).



- **Subtrator Aproximado:** Eddy et al. (Reddy et al., 2018) desenvolveram três versões de subtratores aproximados (*ApproXimate Subtractor* - AXS) que são incorporados em divisores. Cada versão possui uma distância absoluta de erro única e segue um conjunto específico de expressões lógicas. Enquanto as versões um e dois seguem um padrão similar de expressões, a versão três apresenta um conjunto distinto, demonstrando a versatilidade e adaptabilidade do AXS. Os três tipos de células subtratoras aproximadas (AXS1, AXS2 e AXS3) apresentam inovações arquiteturais importantes. O AXSC1, apresentado na Fig. 29, tem célula projetada para reduzir o número de transistores, eliminando um dos circuitos

Figura 28 – Subtrator APXS 2 (Muthulakshmi; Dash; Prabaharan, 2018).



XOR/XNOR presentes em um subtrator exato. A saída de *borrow* (*Bout*) é priorizada sobre a saída de diferença (*D*), permanecendo inalterada para garantir maior precisão nessa parte da operação. O subtrator AXS2, apresentado na Fig. 30, se caracteriza pela combinação das saídas *D* e *Bout*, o que ajuda a reduzir o atraso na operação de subtração. AXS2 oferece versões baseadas em projetos XOR e XNOR, proporcionando flexibilidade no projeto do circuito. O AXSC3, apresentado na Fig. 31, é semelhante ao AXS1 e reduz o número de transistores ao eliminar um dos circuitos XOR/XNOR, mantendo a saída *Bout* constante. Além disso, combina as saídas *D* e *Bout* de maneira similar ao AXS2, otimizando tanto o espaço quanto o tempo de resposta do circuito.

Figura 29 – Subtrator AXS 1 (Reddy et al., 2018).

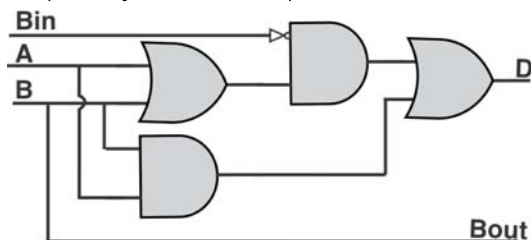


Figura 30 – Subtrator AXS 2 (Reddy et al., 2018).

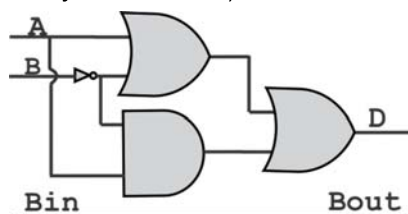
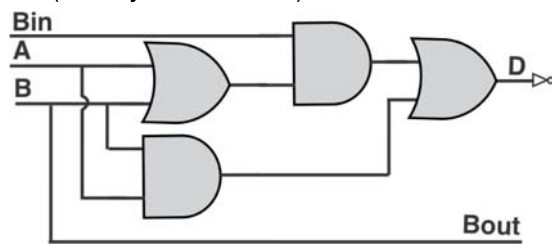


Figura 31 – Subtrator AXS 3 (Reddy et al., 2018).



## 4 APPROXIMATE SUBTRACTOR: UM NOVO SUBTRATOR

No âmbito do processamento de vídeo, destaca-se a escassez de alternativas eficientes para subtratores aproximados. Essa lacuna motivou o desenvolvimento do operador *APProximate Subtractor* (AppS), uma inovação projetada inicialmente para calcular a métrica de distorção SAD. Contudo, o AppS também pode ser aplicado a outras métricas de similaridade e diferentes finalidades. Tradicionalmente, a implementação de subtratores precisos para este fim apresenta desafios significativos, sobretudo devido à elevada demanda computacional, que se torna ainda mais crítica em sistemas embarcados e dispositivos móveis.

O operador AppS emerge como uma solução promissora, almejando melhorar a eficiência computacional. Ele equilibra a precisão necessária no cálculo do SAD com um consumo energético otimizado, aspectos relevantes em aplicações práticas.

A concepção do AppS se baseou na combinação estratégica de portas lógicas. Esse projeto visa atingir um duplo objetivo: minimizar o erro inerente ao cálculo do SAD e, ao mesmo tempo, reduzir tanto o consumo de energia quanto o tempo de processamento. A Figura 32 ilustra o diagrama do subtrator aproximado AppS de um bit. Composto por uma porta XOR e um multiplexador, o AppS gera a saída *S* a partir da diferença dos operandos de entrada *A* e *B*, enquanto a saída de *borrow* é derivada do operando *B* e da entrada *C* (*borrow*). Sua implementação para operandos de entrada de quatro bits também é apresentada na Figura 32, demonstrando a aplicabilidade do AppS em diferentes configurações.

O projeto do AppS é focado em eficiência energética e baixo consumo, mantendo uma precisão aceitável para várias aplicações. A estrutura simplificada, resultante da combinação entre a porta XOR e o multiplexador, facilita a operação de subtração e sua implementação.

Dividido em duas partes, uma precisa e outra imprecisa, o AppS gera os bits mais significativos da subtração e o mínimo de bits significativos necessários, respectivamente. A Figura 33 mostra um exemplo do subtrator aproximado AppS com dois bits aproximados. A seção imprecisa contém dois subtratores AppS para os bits menos significativos (*A0*, *B0* e *borrow*), enquanto a seção precisa utiliza um subtrator exato.

Essa configuração evidencia um equilíbrio eficiente entre precisão e eficiência operacional, essencial para sistemas com restrições de recursos computacionais, como sistemas embarcados ou dispositivos móveis.

$$D = A \oplus B \quad (7)$$

$$B_{out} = \overline{D}.B_{in} + D.B \quad (8)$$

Figura 32 – Subtrator AppS de 1-bit.

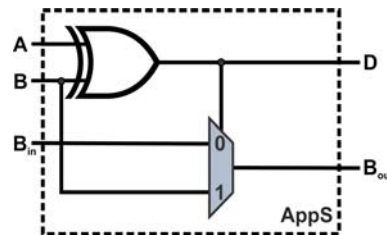
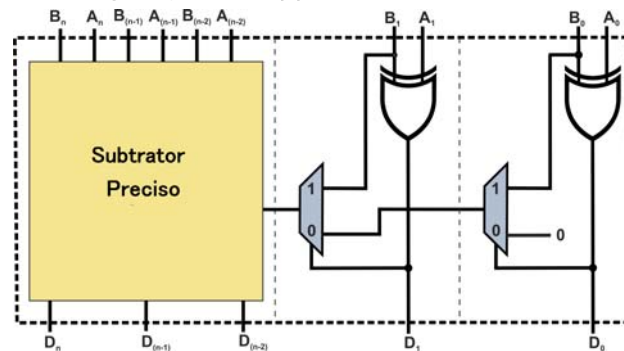


Figura 33 – Exemplo de Configuração do AppS com 4-bits.



A tabela verdade de um subtrator exato e do subtrator AppS estão apresentadas na Tabela 1. Ao examinar as tabelas verdade, é possível constatar que o AppS apresenta a saída D correta em 87,7% das combinações de entrada, enquanto apresenta a saída B\_out (*borrow out*) correta para 75% das combinações de entrada. Ambas as saídas estão corretas simultaneamente em 50% dos casos, nas situações onde A, B e B\_in (*borrow in*) são iguais a (0,0,0), (0,1,0), (1,0,0) e (1,1,0). No total, considerando as duas saídas, o AppS atingiu 68,75% de saídas precisas (11 de 16). As comparações de saídas corretas e incorretas em relação ao subtrator exato foram marcadas com ✓ e ×, respectivamente na Tabela 1.

O uso do subtrator aproximado AppS em sistemas digitais oferece vantagens significativas em termos de eficiência energética e rapidez no processamento, particularmente útil em contextos onde a precisão perfeita não é crucial. A flexibilidade do AppS em se adaptar a operandos com diferentes tamanhos de bits expande sua utilidade, tornando-o uma opção eficaz e adaptável para uma variedade de cenários e exigências de desempenho.

Tabela 1 – Tabela Verdade de um Subtrator Preciso e do AppS

| Entradas |   |      | Preciso |      | AppS |      |
|----------|---|------|---------|------|------|------|
| A        | B | B_in | D       | Bout | D    | Bout |
| 0        | 0 | 0    | 0       | 0    | 0✓   | 0✓   |
| 0        | 0 | 1    | 1       | 1    | 0✗   | 1✓   |
| 0        | 1 | 0    | 1       | 1    | 1✓   | 1✓   |
| 0        | 1 | 1    | 0       | 1    | 1✗   | 1✓   |
| 1        | 0 | 0    | 1       | 0    | 1✓   | 0✓   |
| 1        | 0 | 1    | 0       | 0    | 1✗   | 0✓   |
| 1        | 1 | 0    | 0       | 0    | 0✓   | 0✓   |
| 1        | 1 | 1    | 1       | 1    | 0✗   | 1✓   |

Para mensurar o impacto das configurações de aproximação empregadas no AppS, desenvolveu-se uma análise detalhada, cujos resultados são apresentados nas Tabelas 57, 58 e 59 da sessão B.3 *Máscara de Imprecisão no Apêndice*. Estas tabelas compilam dados de várias máscaras de aproximação utilizadas em subtratores de oito bits, incluindo a designação de cada máscara, a quantidade de subtrações realizadas — totalizando 131.072 operações —, o percentual de cálculos realizados corretamente, e o número de bits aproximados por máscara. Cada máscara consiste em uma sequência de caracteres 'P' (para bits precisos) e 'I' (para bits aproximados), demonstrando como cada configuração afeta a precisão do subtrator. A disposição dos resultados na tabela permite uma visão ordenada do desempenho de cada máscara, de melhor a pior caso, ilustrando claramente como as variações na configuração impactam a eficácia do subtrator.

Os dados revelam que o desempenho do AppS é altamente influenciado pela configuração de aproximação escolhida. Por exemplo, a configuração PPPPPPPI, com sete bits precisos, registrou um sucesso de 100% nas operações, enquanto opções com maior grau de aproximação, como IIIIIIP ou IIIIII, apresentaram apenas 13,35% de precisão.

A análise mais detalhada das tabelas mostra que nas primeiras 19 linhas, que contemplam até três bits aproximados, a taxa de operações corretas oscila entre 50% e 100%. Interessantemente, observa-se que os resultados mais precisos geralmente estão associados aos bits de menor significância. Essa tendência aponta para uma possível correlação entre a imprecisão nos bits menos significativos e a preservação de uma alta taxa de acertos, o que pode ser crucial para aplicar essa tecnologia de forma efetiva em situações práticas.

#### 4.0.1 Comparando AppS com Outros Subtratores

Nesta seção, realizamos uma comparação detalhada entre o subtrator AppS, desenvolvido em nosso trabalho, e outros seis subtratores aproximados estado da arte, incluindo o AXCS de Chen *et al.* (2015), em suas três variantes, e o AXS de Manikantta Reddy *et al.* (2019), também em três versões. Essa seleção foi baseada

em critérios como relevância na literatura e ano de publicação, conforme indicado em (Manikantta Reddy et al., 2019; Chen et al., 2015).

O foco da comparação abrangeu diversos aspectos, como a Tabela Verdade, o consumo de energia e a área do circuito, além de uma análise da precisão, mensurando a porcentagem de operações corretas de cada subtrator em relação a um padrão preciso. Essa análise foi conduzida individualmente para cada subtrator e de forma comparativa entre eles.

A Tabela 2 exibe a tabela verdade dos subtratores, onde os modelos AppS e AXCS2 se destacam. Observa-se que, com o bit Bin a zero, esses subtratores entregam resultados mais precisos. Especificamente, o AppS atinge 100% de precisão com um bit aproximado, e o AXCS2 alcança 75% sob as mesmas condições. Esses desempenhos são evidenciados na Tabela 3, que detalha as subtrações aproximadas de até 8 bits, totalizando 65.536 operações.

A mesma tabela 3 também registra os resultados de um teste exaustivo envolvendo todas as possíveis 65.536 subtrações de 8 bits. Aqui, o subtrator  $AppS_1$  se sobressai ao manter 100% de precisão com um bit aproximado. Com dois bits, a precisão reduz para 75%, e com três, para 56,25%, mostrando robustez mesmo com mais bits aproximados. Com a aproximação de quatro bits, a eficácia diminui para 41,18%, mas esses índices ainda são significativos. Em média, o AppS apresenta o melhor desempenho global, com uma precisão média de 68,11%, destacando-se sobre outros subtratores de ponta. Esta análise confirma a superioridade do AppS em operações de subtração que envolvem bits aproximados.

Tabela 2 – Tabela Verdade dos Subtratores Aproximados

| Entradas |   |                 | Exato |      | AppS |      | AXCS1 |      | AXCS2 |      | AXCS3 |      | AXS1 |      | AXS2 |      | AXS3 |      |
|----------|---|-----------------|-------|------|------|------|-------|------|-------|------|-------|------|------|------|------|------|------|------|
| A        | B | B <sub>in</sub> | D     | Bout | D    | Bout | D     | Bout | D     | Bout | D     | Bout | D    | Bout | D    | Bout | D    | Bout |
| 0        | 0 | 0               | 0     | 0    | 0    | ✓    | 0     | ✓    | 0     | ✓    | 1     | ✗    | 0    | ✓    | 0    | ✓    | 1    | ✗    |
| 0        | 0 | 1               | 1     | 1    | 0    | ✓    | 1     | ✓    | 1     | ✓    | 1     | ✓    | 0    | ✗    | 1    | ✓    | 1    | ✓    |
| 0        | 1 | 0               | 1     | 1    | 1    | ✓    | 1     | ✓    | 1     | ✓    | 1     | ✓    | 1    | ✓    | 0    | ✗    | 1    | ✓    |
| 0        | 1 | 1               | 0     | 1    | 1    | ✗    | 1     | ✗    | 1     | ✓    | 1     | ✓    | 1    | ✓    | 0    | ✗    | 1    | ✓    |
| 1        | 0 | 0               | 1     | 0    | 1    | ✓    | 0     | ✓    | 1     | ✓    | 0     | ✓    | 1    | ✓    | 1    | ✓    | 0    | ✓    |
| 1        | 0 | 1               | 0     | 0    | 1    | ✗    | 0     | ✓    | 1     | ✗    | 0     | ✓    | 1    | ✓    | 0    | ✓    | 1    | ✓    |
| 1        | 1 | 0               | 0     | 0    | 0    | ✓    | 0     | ✓    | 1     | ✗    | 0     | ✓    | 0    | ✓    | 1    | ✗    | 1    | ✗    |
| 1        | 1 | 1               | 1     | 1    | 0    | ✗    | 1     | ✓    | 1     | ✓    | 1     | ✓    | 1    | ✓    | 1    | ✓    | 0    | ✗    |

Tabela 3 – Precisão de Subtrações

| Aproximado<br>bits | Operador |          |          |          |         |         |         |
|--------------------|----------|----------|----------|----------|---------|---------|---------|
|                    | AppS(%)  | AXSC1(%) | AXSC2(%) | AXSC3(%) | AXS1(%) | AXS2(%) | AXS3(%) |
| 1                  | 100      | 50       | 75       | 50       | 50      | 50      | 75      |
| 2                  | 75       | 25       | 56,25    | 25       | 31,25   | 25      | 56,25   |
| 3                  | 56,25    | 12,50    | 42,18    | 12,50    | 18,75   | 12,50   | 42,18   |
| 4                  | 41,18    | 31,64    | 31,64    | 6,25     | 6,25    | 11,32   | 31,34   |
| Média              | 68,11    | 29,79    | 51,27    | 23,44    | 26,56   | 24,71   | 51,19   |

Para configurar nosso experimento de síntese de circuito lógico, utilizamos a ferramenta Incisive da Cadence, versão 9.20, juntamente com o RTL *Compiler*, também da Cadence, na versão 11.10. Empregamos gates da TSMC de 40 nanômetros a 0,9 Volts. Os subtratores foram descritos em Verilog e submetidos a testes com os quatro níveis de aproximação usando a tabela verdade, o que resultou na execução

Tabela 4 – Resultados da Síntese dos Subtratores

| Subtrator                 | Potência $\mu W$ |             |           | Total | Gates  |             |           | Total | Área $nm^2$ |             |           |
|---------------------------|------------------|-------------|-----------|-------|--------|-------------|-----------|-------|-------------|-------------|-----------|
|                           | Total            | Redução (%) | Média (%) |       | Total  | Redução (%) | Média (%) |       | Total       | Redução (%) | Média (%) |
| Exato                     | 1,326            |             |           | 68    |        |             |           | 407   |             |             |           |
| <i>AppS</i> <sub>1</sub>  | 0,499            | -62,37      | -64,39    | 61    | -10,29 | -15,81      |           | 365   | -10,32      | -16,09      |           |
| <i>AppS</i> <sub>2</sub>  | 0,493            | -62,82      |           | 60    | -11,76 |             |           | 360   | -11,55      |             |           |
| <i>AppS</i> <sub>3</sub>  | 0,487            | -63,27      |           | 59    | -13,24 |             |           | 355   | -12,78      |             |           |
| <i>AppS</i> <sub>4</sub>  | 0,41             | -69,08      |           | 49    | -27,94 |             |           | 286   | -29,73      |             |           |
| <i>AXCS</i> <sub>11</sub> | 0,85             | -35,90      | -43,57    | 67    | -1,47  | -5,88       |           | 398   | -2,21       | -9,28       |           |
| <i>AXCS</i> <sub>12</sub> | 0,777            | -41,40      |           | 65    | -4,41  |             |           | 374   | -8,11       |             |           |
| <i>AXCS</i> <sub>13</sub> | 0,716            | -46,00      |           | 63    | -7,35  |             |           | 359   | -11,79      |             |           |
| <i>AXCS</i> <sub>14</sub> | 0,65             | -50,98      |           | 61    | -10,29 |             |           | 346   | -14,99      |             |           |
| <i>AXCS</i> <sub>21</sub> | 1,143            | -13,80      | -31,88    | 65    | -4,41  | -10,66      |           | 385   | -5,41       | -13,33      |           |
| <i>AXCS</i> <sub>22</sub> | 0,95             | -28,36      |           | 62    | -8,82  |             |           | 359   | -11,79      |             |           |
| <i>AXCS</i> <sub>23</sub> | 0,81             | -38,91      |           | 59    | -13,24 |             |           | 339   | -16,71      |             |           |
| <i>AXCS</i> <sub>24</sub> | 0,71             | -46,46      |           | 57    | -16,18 |             |           | 328   | -19,41      |             |           |
| <i>AXCS</i> <sub>31</sub> | 0,852            | -35,75      | -47,42    | 65    | -4,41  | -8,82       |           | 386   | -5,16       | -14,00      |           |
| <i>AXCS</i> <sub>32</sub> | 0,711            | -46,38      |           | 63    | -7,35  |             |           | 357   | -12,29      |             |           |
| <i>AXCS</i> <sub>33</sub> | 0,63             | -52,49      |           | 61    | -10,29 |             |           | 336   | -17,44      |             |           |
| <i>AXCS</i> <sub>34</sub> | 0,596            | -55,05      |           | 59    | -13,24 |             |           | 321   | -21,13      |             |           |
| <i>AXS</i> <sub>11</sub>  | 1,248            | -5,88       | -22,04    | 65    | -4,41  | -6,62       |           | 385   | -5,41       | -9,09       |           |
| <i>AXS</i> <sub>12</sub>  | 0,798            | -39,82      |           | 63    | -7,35  |             |           | 373   | -8,35       |             |           |
| <i>AXS</i> <sub>13</sub>  | 1,131            | -14,71      |           | 63    | -7,35  |             |           | 366   | -10,07      |             |           |
| <i>AXS</i> <sub>14</sub>  | 0,958            | -27,75      |           | 63    | -7,35  |             |           | 356   | -12,53      |             |           |
| <i>AXS</i> <sub>21</sub>  | 0,72             | -45,70      | -49,45    | 59    | -13,24 | -18,75      |           | 328   | -19,41      | -21,44      |           |
| <i>AXS</i> <sub>21</sub>  | 0,727            | -45,17      |           | 56    | -17,65 |             |           | 321   | -21,13      |             |           |
| <i>AXS</i> <sub>23</sub>  | 0,617            | -53,47      |           | 53    | -22,06 |             |           | 315   | -22,60      |             |           |
| <i>AXS</i> <sub>24</sub>  | 0,617            | -53,47      |           | 53    | -22,06 |             |           | 315   | -22,60      |             |           |
| <i>AXS</i> <sub>31</sub>  | 0,704            | -46,91      | -49,21    | 59    | -13,24 | -18,38      |           | 319   | -21,62      | -25,86      |           |
| <i>AXS</i> <sub>32</sub>  | 0,721            | -45,63      |           | 56    | -17,65 |             |           | 306   | -24,82      |             |           |
| <i>AXS</i> <sub>33</sub>  | 0,62             | -53,24      |           | 53    | -22,06 |             |           | 293   | -28,01      |             |           |
| <i>AXS</i> <sub>34</sub>  | 0,649            | -51,06      |           | 54    | -20,59 |             |           | 289   | -28,99      |             |           |

de 65.536 subtrações para cada nível. Definimos por manter os recursos de otimização de circuito em nível baixo para preservar a descrição original dos circuitos, o que permite uma comparação mais justa entre os subtratores analisados. A frequência de operação escolhida foi de 263 MHz, um valor médio que garantiu a eficiência de todos os subtratores testados.

Ao analisar a dissipação de potência, área ocupada e número de *gates* na Tabela 4, é possível identificar diferenças significativas entre as versões dos subtratores. O AppS se destacou positivamente em termos de redução de dissipação de potência e área, apresentando resultados superiores a 62% de redução em todos os casos e superando todos os demais subtratores em todos os níveis de imprecisão. O AppS, com quatro bits aproximados atingiu 69% de redução de dissipação de potência em comparação ao subtrator preciso, enquanto o melhor caso nos subtratores comparados com imprecisão de quatro bits chegou a 55% para o subtrator AXSC3.

Em termos de número de *gates* e de área, o AppS não apresentou o melhor resultado. Considerando o número de *gates*, o AppS ficou atrás dos subtratores AXS2 e AXS3, que atingiram reduções médias de 18,7% e 18,4%, respectivamente, em relação ao subtrator preciso. O AppS atingiu uma redução de 15,8%. Esse resultado, como era esperado, se espelhou para a área ocupada, onde novamente o AppS ocupou a terceira posição, perdendo para os operadores AXS2 e AXS3. Nesse caso, os operadores AXS2 e AXS3 atingiram uma redução de área de 21,4% e 25,9%, respec-



tivamente, enquanto o AppS atingiu uma redução de 16,1%.

A Tabela 5 apresenta os resultados médios obtidos pelos subtratores aproximados analisados. As colunas potência %, *gates* % e Área % indicam as reduções alcançadas em relação ao subtrator preciso, com valores negativos sendo considerados ganho em relação ao subtrator preciso. Por outro lado, a coluna Precisão (%) indica a precisão média dos subtratores, tal qual apresentado na Tabela 3.

Tabela 5 – Sumarização dos Resultados

| <b>Subtrator</b> | <b>Acurácia (%)</b> | <b>Potência (%)</b> | <b>Gates (%)</b> | <b>Área(%)</b> |
|------------------|---------------------|---------------------|------------------|----------------|
| <i>AppS</i>      | 68,11               | -64,39              | -15,81           | -16,09         |
| <i>AXCS1</i>     | 51,27               | -43,57              | -5,88            | -9,28          |
| <i>AXCS2</i>     | 29,79               | -31,88              | -10,66           | -13,33         |
| <i>AXCS3</i>     | 23,44               | -47,42              | -8,82            | -14,00         |
| <i>AXS1</i>      | 26,56               | -22,04              | -6,62            | -9,09          |
| <i>AXS2</i>      | 24,71               | -49,45              | -18,75           | -21,44         |
| <i>AXS3</i>      | 51,19               | -49,21              | -18,38           | -25,86         |

A média mostra que o subtrator AppS alcançou a maior precisão entre os subtratores analisados, com 68,11%. Em termos de redução de dissipação de potência, o AppS também atingiu os melhores médias, com uma diminuição de 64,39% na potência em relação ao subtrator preciso. Por outro lado, o AppS ficou em terceiro lugar na avaliação de número de gates utilizadas e de área ocupada, como já discutido.

## 5 ANÁLISE DO ALGORITMO TZS COM SUBTRATORES APROXIMADOS

Escolhemos o SAD como ponto focal para a aplicação dos subtratores estudados anteriormente, dada sua importância crítica na avaliação da qualidade de vídeo. Modificações no software de referência do codificador VVC, o VTM, na versão 15 (Feldmann, 2020), conforme referenciado em (Team, 2023), possibilitaram uma integração eficaz dos subtratores aproximados com o algoritmo TZS e o SAD usado neste algoritmo. Essa integração destaca a adaptabilidade dos subtratores em diferentes contextos, prometendo avanços importantes em termos de eficiência energética e redução de área, e mantendo a qualidade do vídeo em níveis satisfatórios.

No contexto do processamento de vídeo, nosso estudo exploratório utiliza duas sequências de vídeo distintas, selecionadas pelas suas características. A análise foi particularmente direcionada para blocos de dimensões 16x16, 32x32, 64x64 e 128x128 pixels. Um destaque especial foi dado aos blocos de 32x32 pixels, considerando-os uma dimensão fundamental. Com isso, os experimentos com blocos de 32x32 foram aprofundados, incluindo uma seleção mais ampla de vídeos. A razão para essa ênfase específica nos blocos de 32x32 será detalhadamente explorada e justificada no decorrer do estudo.

Enfatizamos a relevância dos Índices Temporal (*Temporal Index* - TI) e Espacial (*Spatial Index* - SI) na avaliação da complexidade das sequências de vídeo, conforme descrito por Siqueira *et al.* (2020), Cassa *et al.* (2012) e (Siqueira; Correa; Grellert, 2020; Cassa; Naccari; Pereira, 2012). Esses índices são adequados para compreender a dinâmica das sequências e avaliar as técnicas de codificação e processamento de vídeo com precisão. A Tabela 6 apresenta os índices TI e SI para um conjunto de sequências de vídeo. Especificamente, as sequências BQTerrace, com alto TI e SI moderado, e Cactus, com alto SI e TI moderado, foram escolhidas para testar a eficácia das técnicas em cenários variados. Ambas as sequências possuem resolução de 1920x1080 pixels.

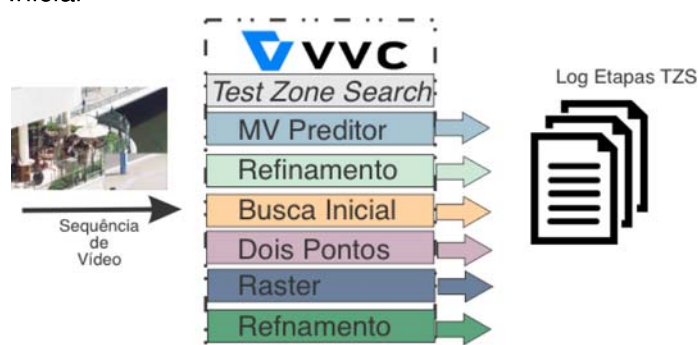
Realizamos experimentos com quatro diferentes modelos de subtratores aproximados - AppS, AXCS1, AXCS2 e AXCS3 - em quatro níveis de aproximação, resultando

Tabela 6 – SI e TI dos Vídeos Analisados, Considerando os 16 Primeiros Frames

| Resolução | Vídeo           | T.I    | S.I.      | FPS |
|-----------|-----------------|--------|-----------|-----|
| 480       | BasketballDrill | 6,913  | 5970,540  | 50  |
| 480       | BQMall          | 5,887  | 11906,350 | 60  |
| 480       | PartyScene      | 7,271  | 14718,577 | 50  |
| 480       | RaceHorsesC     | 21,905 | 8987,960  | 30  |
| 720       | FourPeople      | 2,220  | 6602,646  | 60  |
| 720       | Johnny          | 2,295  | 3942,258  | 60  |
| 720       | KristenAndSara  | 2,148  | 7247,037  | 60  |
| 720       | SlideEditing    | 8,414  | 57639,042 | 30  |
| 720       | SlideShow       | 32,545 | 9027,841  | 20  |
| 1080      | ArenaOfValor    | 7,645  | 7518,191  | 60  |
| 1080      | BasketballDrive | 12,488 | 3433,096  | 50  |
| 1080      | BQTerrace       | 6,056  | 15470,666 | 60  |
| 1080      | Cactus          | 6,886  | 5411,983  | 50  |
| 1080      | MarketPlace     | 28,965 | 5977,375  | 60  |
| 1080      | RitualDance     | 28,507 | 5190,410  | 60  |
| 2160      | Campfire        | 34,317 | 5974,239  | 30  |
| 2160      | CatRobot        | 5,055  | 1962,730  | 60  |
| 2160      | DaylightRoad    | 10,201 | 1592,399  | 60  |
| 2160      | FoodMarket      | 5,552  | 202,701   | 60  |
| 2160      | ParkRunning     | 13,248 | 2834,958  | 50  |
| 2160      | Tango2          | 10,257 | 503,306   | 60  |

em 16 configurações distintas. A escolha desses modelos foi pautada em critérios como precisão e eficiência energética, aspectos que são detalhados no Capítulo 4. A Figura 34 apresenta o esquema utilizado na análise inicial do algoritmo TZS, destacando a interação entre o SAD e os subtratores aproximados e revelando o comportamento específico de cada etapa do TZS. Para uma abrangência ampla, conduzimos 128 execuções experimentais, contemplando variáveis, como os tipos de subtratores e seus níveis de aproximação. Foram usadas as sequências de vídeo e os diferentes Parâmetros de Quantização (QPs 22, 27, 32 e 37) definidos pelas Condições Comum de Teste (*Common Test Conditions* - CTC) do VVC (Bossen, 2013). Foram considerando 16 quadros por sequência, alinhados ao tamanho do grupo de imagens (*Group Of Pictures* - GOP), conforme citado em (Christian, 2023). As execuções realizadas com o subtrator exato (*baseline*) foram utilizadas como referência para comparações diretas.

Figura 34 – Análise Inicial



No estudo da codificação de vídeo utilizando o padrão VVC, o algoritmo TZS é explorado em cinco etapas distintas para uma análise compreensiva:

- Na primeira etapa, estabelecemos um ponto de referência crucial para a nossa análise. Essa fase é dedicada a apresentar o volume de SADs executadas, introduzindo os resultados do subtrator exato como base para comparações futuras.
- Na segunda etapa de nossa investigação, voltamos nossa atenção para um estudo detalhado da relação entre a contagem de SADs nas diversas fases do TZS. Esse foco nos permite comparar o desempenho entre o subtrator exato e o aproximado. Para facilitar a compreensão dos dados coletados, optamos por uma apresentação quantitativa dos resultados, expressando-os em termos percentuais. Esta abordagem simplifica significativamente a interpretação dos dados, permitindo-nos identificar com clareza áreas de melhoria ou retrocesso no número de SADs necessários. Neste contexto, é fundamental destacar a metodologia empregada para avaliar a eficácia dos subtratores. *A eficiência é mensurada pela variação percentual no número de operações de SAD, onde valores negativos representam uma diminuição desejável. Essa redução não se limita apenas a diminuir a quantidade de cálculos exigidos pelo algoritmo TZS; ela também reflete um aumento na eficiência do processamento.* Curiosamente, os subtratores aproximados contribuem para essa melhoria. Eles conseguem manter ou até mesmo elevar a precisão do algoritmo TZS, ao mesmo tempo em que demandam menos recursos computacionais. A importância dos valores negativos encontrados em nossa análise não pode ser subestimada. Eles sinalizam uma otimização significativa no processo, onde a redução na contagem de SADs indica uma notável melhora na performance. O subtrator aproximado, ao realizar a mesma tarefa que o exato com um número menor de operações, não apenas aumenta a velocidade e a eficiência do processamento, mas também pode contribuir para a redução do consumo de energia. Este último aspecto é particularmente crítico em dispositivos móveis e em cenários que exigem processamento intensivo, destacando a relevância de nossas descobertas.
- A terceira etapa do nosso estudo mergulha na comparação direta dos resultados oriundos do cálculo de SAD, focando na média das diferenças absolutas entre o desempenho do subtrator exato e do aproximado. Este momento é crucial para a compreensão aprofundada das nuances e do impacto real que os subtratores aproximados exercem sobre o desempenho geral do processo. Ao examinar as diferenças absolutas médias entre os subtratores, ganhamos percepções valiosas que vão além da simples eficácia. Esta análise nos possibilita quantificar a precisão com que os subtratores aproximados podem substituir o exato, preser-

vando ou até mesmo aprimorando a performance do processo de codificação de vídeo.

- Avançando para a quarta etapa, nosso foco se volta para os efeitos dos subtratores aproximados na precisão dos vetores de movimento. Esta análise é essencial para entender como tais aproximações influenciam a exatidão dos vetores de movimento, um elemento crucial na avaliação da eficiência dos subtratores. Observamos a média da diferença absoluta entre o subtrator exato, utilizado como referência, e cada subtrator aproximado sob análise, revelando as sutilezas desta interação. Importante ressaltar, nas fases três e quatro do processo, o uso de *clusters* torna-se crucial. Essas etapas empregam o recurso de *clustering* para aprimorar o entendimento do comportamento dos subtratores aproximados. A utilização de *clusters* nessas etapas permite uma análise mais detalhada e uma melhor interpretação dos dados, facilitando a identificação de características comuns e anomalias dentro dos grupos formados. Portanto, o *clustering* não apenas simplifica a visualização dos dados, mas também enriquece a análise, proporcionando percepções mais profundas e fundamentadas sobre o conjunto de dados em estudo.
- Finalizando com a quinta etapa, procedemos a avaliação da performance de codificação, o resultado último da atuação dos subtratores aproximados em um cenário real. Este aspecto é crucial para o desempenho global do sistema ao adotar subtratores aproximados. Compreensões consolidadas nesta fase sintetizam os impactos gerais das inovações propostas, fornecendo uma visão holística do desempenho do sistema em termos de codificação de vídeo. Este percurso analítico não só esclarece o papel vital dos subtratores aproximados na otimização do processo de codificação, mas também destaca a importância de uma avaliação estatística rigorosa para validar a eficácia e eficiência das técnicas empregadas.

Durante a execução do TZS, procedemos a coleta de uma vasta gama de dados, abrangendo tanto entradas específicas quanto métricas relevantes. Estes dados foram devidamente armazenados em arquivos de log para análises. A nossa investigação avaliou minuciosamente a eficácia dos subtratores ao longo de diversas fases do TZS, considerando as sequências de vídeo e os quatro QPs definidos pela CTC do VVC. Esta análise abrangente permitiu-nos examinar tanto a frequência de execução do TZS quanto o desempenho dos subtratores aproximados, comparando-os com a versão *baseline* em diferentes tamanhos de blocos.

Aprofundando na investigação sobre os impactos de uso de subtratores aproximados no TZS, nosso estudo detalhou as diversas etapas envolvidas, desde o Preditor

Mediano até a etapa de Varredura, focalizando na avaliação das demandas computacionais e na eficiência desses processos. Especial atenção foi dada aos vídeos BQTerrace e Cactus, sobre os quais realizamos um exame detalhado. Esta análise comparativa entre subtratores precisos e aproximados destacou significativamente a eficiência de codificação alcançada, evidenciando os benefícios potenciais destas tecnologias em aplicações práticas.

Em síntese, este capítulo oferece uma análise detalhada e criteriosa da codificação de vídeo sob a perspectiva do VVC, enfatizando a importância da seleção criteriosa dos tamanhos de blocos, da eficácia dos subtratores aproximados e, sobretudo, da eficiência de codificação. Através deste exame minucioso, visamos contribuir para o avanço do conhecimento nesta área, apresentando conclusões que podem auxiliar no desenvolvimento de soluções de codificação de vídeo mais eficientes e eficazes.

## 5.1 Escolha do Tamanho de Blocos

O conjunto de experimentos apresentados nesta seção concentra-se na análise do uso de tamanhos de bloco distintos na codificação de vídeos quando são usados subtratores aproximados.

A Tabela 7 detalha um experimento inicial, apresentando a escolha dos tamanhos de blocos durante a execução do SAD com o subtrator *baseline*. Ela evidencia a diversidade no uso de diferentes tamanhos de blocos durante a codificação dos vídeos BQTerrace e Cactus.

O experimento revelou uma ampla variedade nos tamanhos de blocos utilizados, variando dos maiores (128x128) aos menores (4x16). Essa diversidade é essencial para melhorar a qualidade e eficiência da compressão de vídeo, adaptando-se às características visuais distintas de cada cena. A utilização variada de tamanhos de blocos facilita uma adaptação mais acurada à complexidade espacial e temporal das cenas, otimizando a precisão e a eficiência da codificação.

Uma análise detalhada da Tabela 7 mostra uma tendência à predominância de blocos pequenos, especialmente de 8x8, que apresentam as maiores contagens. Por exemplo, no vídeo BQTerrace, o bloco de 8x8 foi empregado 4.787.583 vezes, enquanto em Cactus o número foi de 3.081.920. Isso indica que, para a resolução de 1920x1080, mesmo para vídeos com TI e SI divergentes, o codificador apresenta uma preferência no uso de blocos menores, provavelmente pela sua capacidade em representar detalhes precisos e movimentos complexos. Blocos de tamanhos médios, como 32x32 e 64x64, também são significativamente utilizados, refletindo sua utilidade em áreas com menor variação visual. Naturalmente, estes valores mudariam se vídeos de resoluções diferentes fossem analisados.

As diferenças na escolha de tamanhos de blocos entre os vídeos são notáveis. No

Tabela 7 – Contagem de Blocos Usados na Codificação dos Vídeos *BQTerrace* e *Cactus* Usando o Subtrator *Baseline*

| Blocos  | BQTerrace | Cactus    | Total     |
|---------|-----------|-----------|-----------|
| 128x128 | 29.417    | 31.631    | 61.047    |
| 64x64   | 198.250   | 220.083   | 418.333   |
| 32x32   | 791.743   | 914.354   | 1.706.097 |
| 16x16   | 1.916.619 | 1.961.581 | 3.878.200 |
| 8x8     | 4.787.583 | 3.081.920 | 7.869.503 |
| 128x64  | 42.039    | 46.111    | 88.150    |
| 64x128  | 42.202    | 46.419    | 88.621    |
| 64x32   | 303.119   | 374.241   | 677.360   |
| 64x16   | 416.447   | 512.785   | 929.232   |
| 64x8    | 321.805   | 457.739   | 779.544   |
| 64x4    | 249.571   | 385.259   | 634.830   |
| 32x64   | 299.134   | 371.162   | 670.296   |
| 32x16   | 1.167.045 | 1.395.140 | 2.562.185 |
| 32x8    | 1.236.324 | 1.401.677 | 2.638.001 |
| 32x16   | 1.167.045 | 1.395.140 | 2.562.185 |
| 32x8    | 1.236.324 | 1.401.677 | 2.638.001 |
| 16x64   | 412.018   | 504.829   | 916.847   |
| 16x8    | 3.534.400 | 2.856.287 | 6.390.687 |
| 16x4    | 3.038.623 | 2.094.029 | 5.132.651 |
| 8x64    | 351.002   | 449.546   | 800.548   |
| 8x32    | 1.268.912 | 1.395.824 | 2.664.735 |
| 8x16    | 3.571.009 | 2.839.591 | 6.410.600 |
| 6x32    | 1.193.940 | 1.375.901 | 2.569.840 |
| 4x64    | 310.312   | 400.735   | 711.047   |
| 4x32    | 948.458   | 844.436   | 1.792.893 |
| 4x16    | 3.211.678 | 2.250.241 | 5.461.919 |

vídeo BQTerrace os blocos menores são mais usados do que no vídeo Cactus que, por sua vez, tende a usar mais os blocos de tamanho maior. Essa variação sugere que diferenças nas características visuais dos vídeos, como textura, movimento e detalhes, influenciam a seleção dos tamanhos dos blocos durante a codificação.

Adicionalmente, a variedade de uso de blocos retangulares, como 128x64 e 64x32, destaca a flexibilidade do codificador em se adaptar à natureza variada dos dados de vídeo. Por exemplo, o bloco 64x32 foi usado 303.119 vezes em BQTerrace e 374.241 vezes em Cactus, sublinhando sua importância no processo de codificação. As variações nos tamanhos e formatos dos blocos são indicativas da natureza complexa da codificação de vídeo e da necessidade de adaptabilidade para uma compressão eficiente e de alta qualidade.

Com base nos resultados apresentados na Tabela 7, foram avaliados os impactos nas escolhas de tamanhos de bloco quando são usados subtratores aproximados no cálculo do SAD. Os resultados deste experimento estão apresentados na Tabela 8. Cada subtrator foi comparado com o *baseline*, permitindo uma avaliação precisa do desempenho de cada um. Os resultados dessa comparação são expressos em termos percentuais (%), obtidos pela relação entre o desempenho do subtrator aproximado e o do subtrator exato. Essa metodologia facilita a compreensão direta e concisa

dos impactos de cada subtrator, onde valores negativos indicam uma redução no uso daquele tamanho de bloco em comparação ao *baseline*, enquanto valores positivos sinalizam um aumento no uso daquele tamanho de bloco.

Como é possível perceber nos resultados da Tabela 7, o uso de subtratores aproximados no cálculo de SAD teve impacto mínimo no índice da escolha de blocos de tamanhos variados. Nenhum resultado chegou aos 2% de impacto e a imensa maioria ficou abaixo dos 0,5% de impacto. Os sinais dos resultados foram variados, o que indica que houve crescimento ou redução no número de blocos de cada tamanho usados, mas sem um viés que indique com clareza alguma mudança de comportamento no codificador.

Em relação às quatro versões do AppS, é possível perceber que a inclusão de aproximação amplia levemente o uso de blocos de tamanho maior, reduzindo o uso de blocos de tamanho menor. Quanto maior o nível de imprecisão, maior é esta tendência. O mesmo comportamento não é observado para os demais subtratores avaliados.

### 5.1.1 Representatividade

Nesta seção, realizamos uma análise sobre a representatividade do tamanho dos blocos na codificação de vídeo, com ênfase nos vídeos Cactus e BQTerrace, já mencionados anteriormente. A representatividade é dada pela área (em número de amostras) codificada utilizando um dado tamanho de bloco. Nosso objetivo é fundamentar a seleção dos blocos de 32x32 pixels como foco principal para análises mais detalhadas. Essa dimensão foi escolhida após uma avaliação metódica, considerando as características específicas e os resultados obtidos com esses vídeos.

A representatividade  $Rep$  dos diferentes tamanhos de blocos na codificação de vídeo é explicada pela Equação 9. Nessa equação,  $LBloco$  e  $ABloco$  representam a largura e altura do bloco, respectivamente, e  $Ocorr$  denota a frequência de seleção de cada tamanho de bloco. A fórmula é expressa como:

$$Rep = (LBloco \times ABloco) \times Ocorr \quad (9)$$

Em nossa avaliação, consideramos os tamanhos de blocos quadráticos, variando de 128x128 a 8x8 e os resultados estão apresentados na Tabela 9. A área de cada bloco, calculada como o produto de  $LBloco$  e  $ABloco$ , mostra uma variação significativa. Blocos maiores, como os de 128x128, possuem uma área proporcionalmente grande, enquanto os menores, de 8x8, têm uma área relativamente pequena. A média de ocorrências ( $Ocorr.Mdia$ ) sugere uma tendência para blocos menores, com os de 8x8 apresentando uma frequência de uso muito alta. A representatividade ( $Rep$ ) de cada tamanho de bloco, obtida ao multiplicar sua área pela média de ocorrências, fornece uma medida da sua relevância na codificação total, ou seja, indica quantas



amostras do vídeo foram codificadas com aquele tamanho de bloco. Nessa análise, os blocos de 64x64 e 32x32 emergem como os mais significativos.

Observamos que os blocos com maior representatividade são de tamanho 32x32 e 64x64, sendo usados para processar, respectivamente, 29,33% e 28,79% das amostras do vídeo. Com base nessa análise, optamos por focar nos blocos de 32x32 para a maioria dos experimentos a serem apresentados nos capítulos seguintes, uma vez que não seria viável considerar todos os tamanhos de blocos nas análises que se seguem.

## 5.2 Análise de Sensibilidade à Aproximação

Nesta sessão, analisaremos todos os tamanhos de bloco sem interferir na escolha dos blocos pelo codificador. Em seguida, será apresentada a análise específica para blocos de 32x32, onde houve interferência direta na seleção do tamanho de bloco no codificador. A escolha de trabalhar com blocos quadrados visou limitar o escopo do estudo, além de facilitar a implementação na arquitetura de hardware. As análises para blocos de 16x16, 64x64 e 128x128 serão detalhadas no Apêndice - B-Análise para Blocos de Tamanhos Fixo.

### 5.2.1 Contagem de SAD executados: Todos os Blocos

Para entender melhor o desempenho do algoritmo de busca Test Zone Search (TZS), foram realizados experimentos com as sequências de vídeo BQTerrace e Cactus, focando na análise da quantidade de cálculos de Soma das Diferenças Absolutas (SAD) utilizadas em cada etapa do algoritmo. Os resultados desses experimentos foram organizados na Tabela 10, que serve como parâmetro inicial para análises subsequentes e mostra a quantidade de SAD executados com o subtrator *baseline*.

Os dados indicam que o número de cálculos de SAD é fortemente influenciado pelas características específicas de cada vídeo. Por exemplo, no vídeo BQTerrace, observa-se uma predominância de operações SAD na etapa de Busca Inicial, representando 83,029% do total, em comparação com apenas 61,234% no vídeo Cactus. Isso sugere que aspectos particulares do BQTerrace favorecem uma abordagem mais intensiva nesta fase inicial do algoritmo.

Em contrapartida, o vídeo Cactus demonstra uma maior necessidade de operações nas etapas de Varredura e Busca Refinada (Ref). Nessas fases, 28,893% e 6,724% das operações SAD são realizadas, respectivamente, contrastando com apenas 9,385% e 2,992% no BQTerrace. Essas diferenças ressaltam a importância de uma varredura mais detalhada e de ajustes mais finos no vídeo Cactus, destacando como as características intrínsecas de cada sequência influenciam diretamente a eficácia dos subtratores aplicados.

Além disso, as etapas de Preditor Mediano e Dois Pontos também apresentam variações notáveis entre os dois vídeos. O BQTerrace mostra uma maior eficiência no Preditor Mediano, com 4,215% das operações SAD, enquanto apresenta apenas 0,379% nas operações de Dois Pontos. Por outro lado, o Cactus registra 2,688% das operações no Preditor Mediano e 0,461% em Dois Pontos. Essas estatísticas sublinham como as peculiaridades de cada vídeo são decisivas na seleção das técnicas mais adequadas para cada etapa específica do algoritmo. As diferenças observadas nas etapas de Busca Inicial e Varredura, em particular, sugerem que dependendo da natureza do vídeo, uma etapa pode demandar significativamente mais recursos computacionais que a outra.

O estudo do impacto do uso de subtratores aproximados no número de operações de SAD (Soma das Diferenças Absolutas) em diferentes etapas do algoritmo de busca TZS é crucial para avaliar se as vantagens de redução de área e dissipação de potência desses subtratores superam as possíveis perdas de eficiência. A preocupação principal é que os subtratores aproximados, apesar de trazerem benefícios teóricos, possam alterar as decisões do codificador devido aos resultados aproximados nas operações de SAD, o que poderia aumentar o número dessas operações e neutralizar os ganhos obtidos.

A análise detalhada foi realizada para os vídeos BQTerrace e Cactus (Tabelas 11 e 12), com os resultados compilados nas tabelas referenciadas, ilustrando o desempenho dos subtratores em cada etapa do TZS. No vídeo BQTerrace, o subtrator  $AXCS2_1$  se destacou ao reduzir o número de operações de SAD nas etapas de Preditor Mediano (PM) e Busca Inicial (BI) em 0,17% e 0,19%, respectivamente. Outros subtratores, como o  $AXCS1_1$  e o  $AXCS1_2$ , mostraram melhorias significativas nas etapas de Dois Ponto (DP) e Varredura, com reduções de 0,07% e 0,51%. O subtrator  $AXCS3_1$  foi notável no Busca Refinada (Ref), com uma diminuição de 0,75%. Quando observada a coluna total, o destaque vai para o subtrator  $AXCS1_2$  reduzindo 0,055% e  $AXCS2_1$  reduzindo 0,152%.

Contudo, nem todos os subtratores apresentaram resultados positivos. O  $AppS_2$ , por exemplo, aumentou o número de cálculos de SAD tanto no Preditor Mediano quanto na Busca Inicial, com incrementos de 1,83% e 1,93%, respectivamente. O  $AXCS2_4$  resultou no maior aumento no Dois Ponto, com 53,59%, e elevações também foram observadas na etapa de Varredura e Busca Refinada (Ref).

Para o vídeo Cactus, os padrões foram similares, ressaltando a influência significativa dos subtratores aproximados. O subtrator  $AXCS1_4$  se destacou no Preditor Mediano, com uma redução de 0,49% nas operações de SAD. Enquanto isso, o  $AXCS2_1$  e os subtratores  $AppS_2$  e  $AppS_3$  também obtiveram bons resultados em várias etapas, superando os 0,30% em reduções. Em contrapartida, o  $AXCS2_4$  demonstrou um aumento substancial nas operações em todas as etapas avaliadas, reforçando a

necessidade de uma seleção criteriosa dos subtratores para otimizar o desempenho. Observando a coluna total, apresentam a maior redução o subtrator  $AppS_2$  e  $AppS_3$  reduzindo 0,315 e 0,346, respectivamente.

#### 5.2.1.1 Análise do Resultado do Cálculo do SAD

O próximo experimento avaliou as diferenças médias absolutas entre os resultados de SAD com o uso do subtrator exato e com o uso dos subtratores aproximados. Os resultados estão apresentados na Tabela 13. A tabela apresentada sumariza a diferença média absoluta observada entre todos os vídeos analisados até o momento. Essa avaliação revelou uma diferença média de 1.686,44. Esse valor serve como um patamar de referência essencial para comparar o desempenho individual de cada subtrator. Tal valor demonstra que, em média, os subtratores operam dentro dessa faixa de desempenho. Além disso, um desvio padrão de 1.094,55 foi observado, indicando uma variabilidade significativa entre os resultados, o que ressalta a variação no desempenho dos subtratores.

No que diz respeito ao desempenho individual, o subtrator  $AppS_1$  destacou-se ao atingir um valor de zero, igualando-se ao subtrator de referência, o *baseline*. Por outro lado, o subtrator  $AppS_4$  apresentou o maior valor observado, 4114,42, posicionando-se como o menos preciso.

No geral, o  $AppS$  atingiu resultados intermediários de precisão, com resultados melhores do que o subtrator  $AXCS2$  e com resultados inferiores aos subtratores  $AXCS3$ .

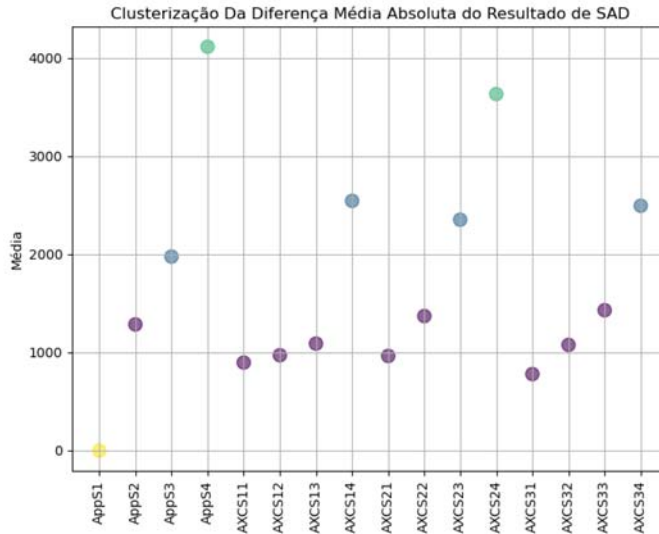
A Figura 37 classifica os subtratores em quatro *clusters* distintos, facilitando a identificação dos que mais se aproximam do resultado exato. Primeiramente, é notável a variação dos valores da média da diferença absoluta, evidenciada pela posição vertical dos pontos no gráfico, que demarcam as discrepâncias observadas.

O *Cluster* Amarelo se destaca por incluir apenas um subtrator, que apresenta a menor média de diferença, sugerindo uma aproximação quase exata ao valor real. Este *cluster* parece ser o menos problemático em termos de precisão. Já os *Clusters* Roxo e Azul englobam a maioria dos subtratores, com médias que variam de baixas a intermediárias, indicando uma variação considerável na qualidade das aproximações dentro destes grupos. O *Cluster* Verde (3) inclui subtratores com as maiores médias, refletindo as maiores discrepâncias e, conseqüentemente, menor precisão.

#### 5.2.1.2 Análise da Distância dos Vetores de Movimento

A análise apresentada nessa seção avalia o desempenho dos subtratores com foco na diferença absoluta entre o vetor de movimento gerado com os subtratores aproximados e o subtrator *baseline*. Os resultados estão apresentados na Tabela 14. A distância média absoluta foi de 0,59, estabelecendo um parâmetro para avaliação do desempenho para comparação. Esse resultado é acompanhado por um desvio

Figura 35 – Classificação dos Subtratores por Resultado do Cálculo de SAD. Todos os Blocos



padrão de 0,655, destacando a variabilidade entre os subtratores.

O subtrator  $AppS_1$  atingiu a média de 0, indicando ausência de impactos negativos e demonstrando uma alta precisão. Em contraste, o  $AXCS2_4$  apresentou uma distância média de 2,23, refletindo menor precisão em suas aproximações.

O subtrator  $AppS$  apresentou desempenho intermediário nesse critério, superando o subtrator  $AXCS2$ , mas ficando atrás dos resultados alcançados pelos subtratores  $AXCS3$ .

Neste caso, os dados foram segmentados em dois *clusters*, apresentados na Figura 36 com diferenças marcantes. O *Cluster 0*, identificado pela cor roxa, apresenta valores inferiores de diferença média absoluta, refletindo aproximações mais próximas ao subtrator *baseline*. O *Cluster 1*, em amarelo, por sua vez, mostra valores mais altos, sinalizando a necessidade de otimização nas aproximações realizadas.

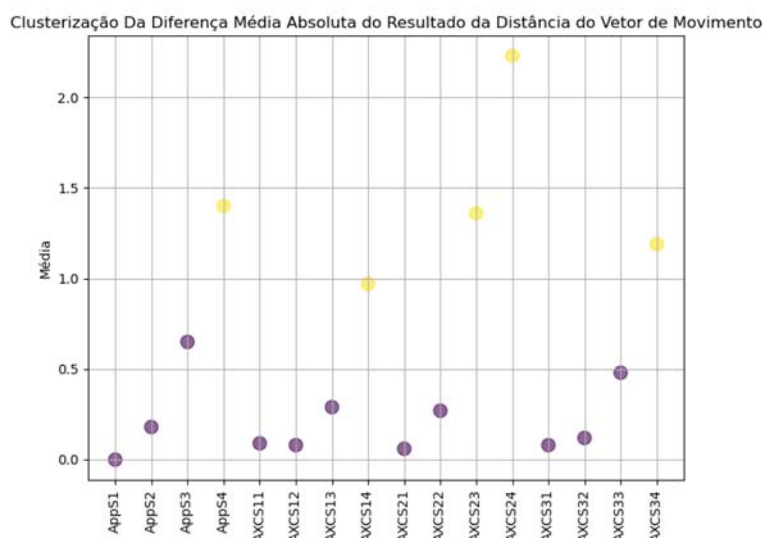
A escala do eixo da média, que oscila entre 0 e 2, evidencia que as diferenças são pequenas, mas suficientes para justificar a clusterização. Análises das características dos subtratores que apresentam as maiores diferenças podem revelar o impacto de variáveis como tipo de movimento e condições operacionais.

### 5.2.1.3 Eficiência de Codificação

O último experimento considerou a eficiência de codificação, em termos de BD-Rate da Tabela 15 que indica, para uma mesma qualidade de vídeo, após uma alteração em relação ao codificador original. Os resultados estão apresentados na Tabela 15, com a eficiência de codificação para os dois vídeos já mencionados, utilizando os diferentes operadores ( $AppS$ ,  $AXCS1$ ,  $AXCS2$  e  $AXSC3$ ) e em quatro níveis de aproximação (1, 2, 3 e 4).

A eficiência de codificação em BD-RATE é uma métrica essencial para avaliar como

Figura 36 – Classificação dos Vetores de Movimento por Subtrator Aproximado. Todos os Tamanhos de Bloco



diferentes configurações e algoritmos de codificação influenciam a taxa de bits necessária para manter uma qualidade de vídeo consistente. Valores negativos nesta métrica indicam uma economia de bits, enquanto valores positivos sugerem um aumento no uso de bits.

Em geral, os subtratores aproximados AXSC1, AXSC2 e AXSC3 mostram uma tendência a economizar bits em comparação com a referência. Isso é evidente pelos valores negativos observados na maioria dos casos. Entretanto, há exceções, como o AXSC3 no vídeo Cactus, que apresenta um valor médio levemente positivo 0,0087%, indicando um pequeno aumento no uso de bits para esse cenário específico.

O AXSC1 se destaca como o mais eficiente em termos de economia de bits na primeira análise, alcançando uma média de -0,0901%. No entanto, seu desempenho mostra-se menos consistente no vídeo Cactus, com uma média de -0,0350%. O AXSC2 exibe uma variação maior entre os vídeos. Enquanto apresenta um desempenho moderadamente bom no primeiro vídeo com -0,0786%, ele mostra uma ligeira piora no Cactus, onde a economia de bits se reduz a 0,0087%. O AXSC3, por outro lado, é o menos consistente, variando de uma pequena economia a um aumento no uso de bits, indicando que sua eficácia pode depender mais do conteúdo do vídeo ou das condições específicas de teste.

Além disso, a análise por sequência revela as diferenças entre as sequências individuais dentro de cada vídeo, demonstrando a sensibilidade dos algoritmos a diferentes tipos de conteúdo. Por exemplo, uma configuração específica do AXSC2 na primeira análise mostra um aumento significativo na economia de bits em comparação com sua média geral. Nos resultados apresentados nas seções anteriores, destacamos as diferenças médias absolutas nos resultados de SAD e nas variações dos vetores de

movimento. Observamos que, apesar da diferença média absoluta no SAD ser considerável, a variação máxima nos vetores de movimento foi de apenas 2 unidades. Esse achado indica que os vetores de movimento, mesmo diante de maiores variações nos valores de SAD, apontam para posições quase idênticas às determinadas pelo subtrator exato. Tal observação sugere que a influência dos subtratores aproximados na precisão dos vetores de movimento, e consequentemente na qualidade perceptível do vídeo, é mínima.

Os subtratores aproximados demonstram uma melhoria na eficiência de codificação, reduzindo o número de operações SAD necessárias sem impactar significativamente a acurácia na estimação de movimento, que é vital para manter a qualidade do vídeo. Essa eficiência aprimorada é fundamental para otimizar o desempenho e o custo de processamento em diversas aplicações de vídeo. A análise do BD-RATE médio total, considerando todos os vídeos e algoritmos, revelou uma modesta economia de bits, reforçando que o uso de subtratores aproximados não apenas melhora a qualidade de codificação mas também contribui significativamente para a redução no número de operações SAD. Tais resultados reforçam a importância dos subtratores aproximados na modernização das técnicas de codificação de vídeo.

### 5.2.2 Análise de Blocos 32x32

Nesta investigação, optou-se por utilizar blocos de tamanho 32x32. Essa escolha foi embasada na maior representatividade que tal dimensão de bloco oferece, como já discutido no capítulo anterior. A adoção dessa abordagem metodológica permitiu uma expansão significativa do escopo da pesquisa, viabilizando a inclusão de um conjunto mais amplo de vídeos para avaliação.

A investigação foi conduzida de forma independente por resolução, considerando resoluções de 480, 720, 1080 e 2160 (4k), garantindo uma diversidade representativa em termos de qualidade e complexidade de imagem. No total, foram selecionados os vinte e um vídeos definidos na CTC (Bossen, 2013), distribuídos entre estas resoluções. Assim, os vídeos escolhidos para essa análise foram: ArenaOfValor, BasketballDrill, BasketballDrive, iBQMall, BQTerrace, Cactus, Campfirebt709videoRange, CatRobotjvet, DaylightRoad, FoodMarket, FourPeople, Johnny, KristenAndSara, Marketplace, ParkRunning, PartyScene, RaceHorsesC, RitualDance, SlideEditing, SlideShow e Tango.

Esta metodologia enriqueceu a qualidade da análise, tornando os resultados mais robustos e informativos. A análise detalhada, apresentada na Tabela 16, examina a performance do subtrator *baseline* através das contagens de SAD nas várias etapas do algoritmo Test Zone Search (TZS). Esta investigação destaca padrões importantes que diferenciam cenários de codificação mais ou menos eficientes, ressaltando a necessidade de identificar os melhores e piores casos para entender as flutuações na

eficiência da codificação.

As etapas consideradas nesta análise incluem o Preditor Mediana (PM), a Busca Inicial (BI), a Busca de Dois Pontos (DP), a Varredura e a Busca Refinada (Ref). Observamos que o aumento das contagens de SAD é proporcional à resolução do vídeo. A análise estatística da Tabela 16 destaca que as etapas de Busca Inicial, Varredura e Busca Refinada são aquelas em que mais SADs são executadas, acumulando a maior parte dos cálculos.

Por exemplo, na resolução de 480p, as contagens de SAD são relativamente menores, mas em UHD 4k podem atingir bilhões, indicando a enorme carga computacional necessária para processar vídeos de alta resolução. Na resolução de UHD 4k, a etapa de BI no pior caso apresentou uma contagem de SAD superior a 2 bilhões, contrastando com cerca de 126 milhões no pior caso para a resolução de 480. Essa tendência é consistente em todas as resoluções, com a BI mostrando sempre os valores mais altos, seguidos pelas etapas de Varredura e Ref.

Além disso, na resolução de UHD 4k, o vídeo Campfirebt709videoRange atingiu uma contagem total de SAD de 8 bilhões, com as etapas de BI e Varredura contribuindo significativamente para esse total. Em contraste, o vídeo CatRobotjvet, também em UHD 4k, teve uma contagem de SAD de 843 milhões, mostrando a variabilidade entre diferentes conteúdos.

Para a resolução de 1080, o vídeo MarketPlace teve 2,329 bilhões na contagem total de SAD, com números elevados especialmente nas etapas de BI, Varredura e Ref, enquanto o vídeo BQTerrace foi o melhor caso, com 89 milhões de operações de SAD, refletindo menor complexidade de movimento e detalhes visuais.

Na resolução de 720, o vídeo SlideEditing teve uma contagem total de 20 milhões de operações de SAD como o pior caso, e o vídeo SlideShow, com 71 milhões de cálculos, mostrou variações notáveis entre os vídeos, principalmente nas etapas de Varredura e BI.

Finalizando com a resolução de 480, o vídeo RaceHorsesC teve a maior contagem de SAD com 126 milhões, e o vídeo BQMall registrou o menor total, com 41 milhões de operações. Esses resultados destacam a variação significativa entre os cenários, com as etapas de BI, Varredura e Ref se destacando nas diferenças observadas.

Os resultados desta análise ressaltam a diversidade nas contagens de SAD entre diferentes vídeos, mesmo dentro da mesma resolução, e destacam as etapas de Busca Inicial, Varredura e Busca Refinada (Ref) como as principais contribuintes para a contagem total de operações de SAD. Isso enfatiza o impacto substancial que o conteúdo do vídeo exerce sobre a contagem de SAD, uma métrica essencial para avaliar a eficácia da compressão e codificação de vídeos em várias resoluções.

Tabela 8 – Contagem de Blocos Utilizando Subtratores Aproximados

| Blocos  | $AppS_1$ | $AppS_2$ | $AppS_3$ | $AppS_4$ | $AXCS1_1$ | $AXCS1_2$ | $AXCS1_3$ | $AXCS1_4$ |
|---------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|
| 128x128 | 0        | 0,103    | 0,159    | 0,261    | 0,226     | 0,209     | 0,104     | 0,308     |
| 64x64   | 0        | 0,421    | 0,656    | 0,783    | 0,485     | 0,435     | 0,423     | 1,013     |
| 32x32   | 0        | 0,374    | -0,449   | 2,436    | -1,883    | 1,297     | 2,447     | 6,851     |
| 16x16   | 0        | -0,517   | -0,1     | -1,081   | 0,129     | -0,36     | 0,123     | 0,36      |
| 8x8     | 0        | -1,135   | -0,987   | -1,104   | -0,445    | -0,347    | -0,574    | 0,168     |
| 128x64  | 0        | -0,154   | 0,585    | 0,188    | 0,88      | 0,404     | -0,25     | 1,037     |
| 64x128  | 0        | -0,196   | 0,816    | 0,019    | 0,989     | 0,508     | -0,17     | 1,062     |
| 64x32   | 0        | 0,483    | 0,638    | 0,414    | 0,697     | 0,498     | 0,645     | 1,296     |
| 64x16   | 0        | -0,103   | -0,028   | 0,258    | -0,055    | 0,06      | -0,236    | 0,896     |
| 32x64   | 0        | 0,457    | 0,634    | 0,715    | 0,495     | 0,352     | 0,546     | 1,422     |
| 32x16   | 0        | -0,643   | -0,228   | -0,972   | -0,035    | 0,049     | 0,258     | 0,781     |
| 32x8    | 0        | -0,555   | -0,433   | -1,296   | 0,074     | 0,001     | 0,151     | 0,421     |
| 16x64   | 0        | -0,072   | 0,377    | 0,764    | -0,169    | 0,206     | 0,158     | 1,116     |
| 16x8    | 0        | -0,678   | -0,622   | -1,5     | 0         | -0,77     | -0,527    | -0,433    |
| 16x4    | 0        | -0,367   | -0,79    | -1,371   | -0,416    | -0,761    | -0,893    | -0,692    |
| 8x64    | 0        | -1,539   | -0,191   | -1,141   | -1,111    | -0,227    | -0,182    | 0,688     |
| 8x32    | 0        | -0,787   | 0,503    | -0,33    | 0,522     | 0,428     | 0,041     | 0,682     |
| 8x16    | 0        | -0,659   | -0,521   | -0,764   | 0,005     | -0,407    | -0,34     | -0,131    |
| 6x32    | 0        | -0,953   | 0,177    | -0,643   | -0,147    | 0,242     | 0,218     | 0,789     |
| 4x64    | 0        | -1,436   | 1,024    | -0,13    | -0,233    | 0,493     | -0,035    | 0,67      |
| 4x32    | 0        | -0,704   | 0,141    | -0,403   | 0,294     | -0,279    | -0,029    | -0,264    |
| 4x16    | 0        | -0,701   | -0,243   | -0,465   | 0,141     | 0,091     | -0,034    | -0,161    |

| Blocos  | $AXCS2_1$ | $AXCS3_2$ | $AXCS2_3$ | $AXCS3_4$ | $AXCS2_1$ | $AXCS3_2$ | $AXCS2_3$ | $AXCS3_4$ |
|---------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 128x128 | 0,178     | 0,326     | 0,321     | 0,672     | 0,109     | 0,195     | 0,225     | 0,461     |
| 64x64   | 0,638     | 0,733     | 0,804     | 1,035     | 0,302     | 0,239     | 0,55      | 0,942     |
| 32x32   | 4,944     | 2,088     | 2,614     | 6,107     | 2,841     | -0,136    | 2,117     | -1,467    |
| 16x16   | -0,563    | -0,886    | -0,372    | 0,547     | -0,363    | -0,626    | 0,075     | 0,971     |
| 8x8     | -0,449    | -0,947    | 0,522     | -0,305    | -1,588    | -1,144    | -0,363    | -0,095    |
| 128x64  | 0,879     | 1,057     | 0,841     | 0,283     | 0,6       | -0,104    | -0,41     | 0,417     |
| 64x128  | 1,244     | 1,087     | 1,187     | 0,393     | 0,509     | -0,09     | 0,061     | 0,4       |
| 64x32   | 1,204     | 0,664     | 0,926     | 1,024     | 0,435     | -0,066    | 1,157     | 1,187     |
| 64x16   | 0,59      | 0,231     | 0,47      | 1,05      | 0,078     | -0,01     | 0,398     | 1,237     |
| 32x64   | 1,211     | 0,685     | 0,814     | 0,935     | 0,492     | -0,1      | 1,019     | 1,261     |
| 32x16   | -0,131    | -0,808    | 0,058     | 0,672     | 0,027     | -0,379    | 0,054     | 0,674     |
| 32x8    | -0,22     | -0,66     | -0,286    | 0,913     | 0,02      | -0,403    | 0,534     | 1,497     |
| 16x64   | 0,896     | 0,463     | 0,585     | 1,158     | 0,587     | -0,012    | 0,625     | 1,752     |
| 16x8    | -0,653    | -1,438    | -0,35     | -0,33     | -1,255    | -1,011    | -0,127    | 0,243     |
| 16x4    | 0,224     | -0,797    | -0,32     | -0,719    | -1,146    | -0,982    | -0,328    | -0,541    |
| 8x64    | 0,278     | -1,281    | -0,489    | -0,158    | -0,149    | -1,011    | -0,044    | 0,711     |
| 8x32    | -0,278    | -0,412    | -0,211    | 1,049     | -0,027    | -0,048    | 0,451     | 1,534     |
| 8x16    | -0,263    | -1,349    | -0,067    | -0,33     | -1,199    | -0,932    | -0,048    | 0,498     |
| 6x32    | 0,159     | -0,924    | -0,057    | 0,588     | 0,214     | -0,468    | 0,057     | 0,911     |
| 4x64    | -0,794    | -1,13     | -1,191    | 0,742     | 0,605     | -0,414    | 0,285     | 1,956     |
| 4x32    | 0,009     | -1,338    | -0,156    | 0,777     | -1,193    | -0,495    | 0,195     | 0,932     |
| 4x16    | 0,38      | -0,975    | 0,506     | 0,328     | -1,317    | -1,211    | 0,03      | 0,791     |



Tabela 9 – Representatividade na Escolha de Blocos.

| Blocos  | LBlocoxABloco | BQTerrace | Cactus    | Ocor.Mediana | Rep         | (%)   |
|---------|---------------|-----------|-----------|--------------|-------------|-------|
| 128x128 | 16.384        | 29.417    | 31.631    | 30.524,00    | 500.105.216 | 16,79 |
| 64x64   | 4096          | 198.250   | 220.083   | 209.166,50   | 856.745.984 | 28,76 |
| 32x32   | 1.024         | 791.743   | 914.354   | 853.048,50   | 873.521.664 | 29,33 |
| 16x16   | 256           | 1.916.619 | 1.961.581 | 1.939.100,00 | 496.409.600 | 16,67 |
| 8x8     | 64            | 4.787.583 | 3.081.920 | 3.934.751,50 | 251.824.096 | 8,45  |

Tabela 10 – Contagem de SAD com Subtrator *Baseline* para todos os Tamanhos de Bloco em cada uma das Etapas do TZS

| Etapas           | BQTerrace      |         | Cactus         |         |
|------------------|----------------|---------|----------------|---------|
|                  | Quantidade     | (%)     | Quantidade     | (%)     |
| Preditor Mediano | 12.685.550     | 4,215   | 10.108.343     | 2,688   |
| Busca Inicial    | 249.859.406    | 83,029  | 230.252.960    | 61,234  |
| Dois Pontos      | 1.140.688      | 0,379   | 1.733.480      | 0,461   |
| Varredura        | 28.242.048     | 9,385   | 108.643.757    | 28,893  |
| Refinada         | 9.004.104      | 2,992   | 25.284.943     | 6,724   |
| Total            | 300.931.796,40 | 100,000 | 376.023.483,00 | 100,000 |

Tabela 11 – Quantidade Percentual de SADs Executados em Comparação com o Subtrator *Baseline*. Valores Negativos Sinalizam uma Redução nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1,570,066.

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | 1,830               | 1,930            | 2,310          | 2,740        | 1,490       | 2,069    |
| <i>AppS</i> <sub>3</sub>  | 0,490               | 0,750            | -0,230         | 5,100        | 2,670       | 1,222    |
| <i>AppS</i> <sub>4</sub>  | 0,630               | 1,520            | 0,860          | 24,180       | 14,230      | 4,014    |
| <i>AXCS</i> <sub>11</sub> | 0,430               | 0,400            | -0,280         | -0,070       | 0,250       | 0,368    |
| <i>AXCS</i> <sub>12</sub> | -0,090              | -0,110           | -0,510         | 0,340        | 0,470       | -0,055   |
| <i>AXCS</i> <sub>13</sub> | 0,280               | 0,380            | -0,010         | 3,160        | 1,420       | 0,678    |
| <i>AXCS</i> <sub>14</sub> | 0,100               | 0,500            | 0,350          | 13,580       | 6,980       | 1,908    |
| <i>AXCS</i> <sub>21</sub> | -0,170              | -0,190           | -0,350         | 0,270        | -0,140      | -0,152   |
| <i>AXCS</i> <sub>22</sub> | 0,460               | 0,580            | 0,590          | 3,630        | 1,450       | 0,907    |
| <i>AXCS</i> <sub>23</sub> | 0,720               | 1,150            | 0,910          | 12,560       | 6,330       | 2,387    |
| <i>AXCS</i> <sub>24</sub> | 0,060               | 1,280            | 0,480          | 53,590       | 24,900      | 6,844    |
| <i>AXCS</i> <sub>31</sub> | -0,040              | -0,030           | -0,270         | 0,690        | -0,750      | 0,013    |
| <i>AXCS</i> <sub>32</sub> | 0,370               | 0,490            | 0,590          | 1,840        | 0,140       | 0,617    |
| <i>AXCS</i> <sub>33</sub> | 0,440               | 0,660            | 0,130          | 3,800        | 1,690       | 0,993    |
| <i>AXCS</i> <sub>34</sub> | 0,350               | 0,930            | -0,180         | 16,970       | 9,260       | 2,671    |

Tabela 12 – Comparação da Quantidade de SADs Executadas: Subtrator *Baseline* vs. Subtrator Aproximado no Vídeo Cactus, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo: Cactus, TI: 6,056 e SI: 1.570,066

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | -0,070              | 0,030            | 0,540          | -0,950       | -0,890      | -0,315   |
| <i>AppS</i> <sub>3</sub>  | -0,050              | 0,350            | 0,530          | -1,470       | -2,040      | -0,346   |
| <i>AppS</i> <sub>4</sub>  | 0,060               | 1,040            | 1,420          | -1,160       | -1,390      | 0,216    |
| <i>AXCS</i> <sub>11</sub> | 0,000               | 0,000            | 0,180          | -0,280       | 0,000       | -0,080   |
| <i>AXCS</i> <sub>12</sub> | 0,370               | 0,420            | 0,730          | 0,090        | 0,480       | 0,329    |
| <i>AXCS</i> <sub>13</sub> | 0,160               | 0,290            | 0,770          | -0,690       | -0,570      | -0,052   |
| <i>AXCS</i> <sub>14</sub> | -0,490              | 0,050            | 1,350          | 1,020        | 0,350       | 0,342    |
| <i>AXCS</i> <sub>21</sub> | 0,000               | -0,040           | -0,280         | -0,060       | -0,080      | -0,048   |
| <i>AXCS</i> <sub>22</sub> | 0,230               | 0,480            | 1,250          | -0,100       | -0,350      | 0,253    |
| <i>AXCS</i> <sub>23</sub> | 0,160               | 0,730            | 1,420          | 0,550        | -0,430      | 0,588    |
| <i>AXCS</i> <sub>24</sub> | 0,160               | 1,640            | 2,270          | 6,140        | 3,870       | 3,053    |
| <i>AXCS</i> <sub>31</sub> | 0,310               | 0,290            | 0,270          | -0,330       | -0,490      | 0,059    |
| <i>AXCS</i> <sub>32</sub> | 0,370               | 0,410            | 0,430          | -0,040       | -0,720      | 0,203    |
| <i>AXCS</i> <sub>33</sub> | 0,370               | 0,650            | 1,080          | -0,930       | -1,830      | 0,021    |
| <i>AXCS</i> <sub>34</sub> | -0,060              | 0,690            | 0,860          | -1,140       | -1,500      | -0,005   |

Tabela 13 – Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator *Baseline* e Subtratores Aproximados para Todos os Tamanhos de Blocos

| Subtrator | Média     | Subtrator | Média     |
|-----------|-----------|-----------|-----------|
| $AppS_1$  | 0,000     | $AXCS1_1$ | 897,250   |
| $AppS_2$  | 1.285,600 | $AXCS1_2$ | 971,840   |
| $AppS_3$  | 1.976,460 | $AXCS1_3$ | 1.090,450 |
| $AppS_4$  | 4.114,420 | $AXCS1_4$ | 2.544,520 |
| Subtrator | Média     | Subtrator | Média     |
| $AXCS2_1$ | 964,770   | $AXCS3_1$ | 779,970   |
| $AXCS2_2$ | 1.370,960 | $AXCS3_2$ | 1.077,140 |
| $AXCS2_3$ | 2.352,480 | $AXCS3_3$ | 1.430,340 |
| $AXCS2_4$ | 3.632,230 | $AXCS3_4$ | 2.494,680 |

Tabela 14 – Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator *Baseline* e Subtratores Aproximados para Todos os Tamanhos de Blocos

| Subtrator | Média | Subtrator | Média |
|-----------|-------|-----------|-------|
| $AppS_1$  | 0,000 | $AXCS1_1$ | 0,090 |
| $AppS_2$  | 0,180 | $AXCS1_2$ | 0,080 |
| $AppS_3$  | 0,650 | $AXCS1_3$ | 0,290 |
| $AppS_4$  | 1,400 | $AXCS1_4$ | 0,970 |
| Subtrator | Média | Subtrator | Média |
| $AXCS2_1$ | 0,060 | $AXCS3_1$ | 0,080 |
| $AXCS2_2$ | 0,270 | $AXCS3_2$ | 0,120 |
| $AXCS2_3$ | 1,360 | $AXCS3_3$ | 0,480 |
| $AXCS2_4$ | 2,230 | $AXCS3_4$ | 1,190 |

Tabela 15 – Resultados de BD-RATE. Blocos de Todos os Tamanhos

| Vídeo: BQTerrace |         |         |         |         |
|------------------|---------|---------|---------|---------|
|                  | AppS    | AXSC1   | AXSC2   | AXSC3   |
| 1                | 0,000   | -0,1221 | -0,125  | -0,0298 |
| 2                | -0,1051 | -0,1194 | -0,0569 | -0,120  |
| 3                | -0,0637 | -0,1024 | -0,0705 | -0,0666 |
| 4                | -0,0299 | -0,0167 | -0,0621 | -0,0645 |
| Vídeo: Cactus    |         |         |         |         |
|                  | AppS    | AXSC1   | AXSC2   | AXSC3   |
| 1                | 0,000   | -0,0061 | 0,0048  | 0,0025  |
| 2                | 0,0106  | 0,0098  | -0,0263 | 0,015   |
| 3                | 0,1164  | -0,0747 | -0,0122 | -0,028  |
| 4                | 0,0781  | -0,0693 | 0,0687  | -0,019  |

Tabela 16 – Contagem de SAD com Subtrator Baseline para Tamanhos de Bloco 32x32 em Cada uma das Etapas do TZS

| Resolução | Vídeo                   | PM                 | PM(%)        | BI                   | BI(%)         | DP                | DP(%)        | Varredura             | Varredura(%)  | Refinada             | Refinada(%)   | Total                 |
|-----------|-------------------------|--------------------|--------------|----------------------|---------------|-------------------|--------------|-----------------------|---------------|----------------------|---------------|-----------------------|
| 480       | BasketballDrill         | 1.527.257          | 2,447        | 34.938.106           | 55,974        | 235.630           | 0,377        | 20.770.603            | 33,276        | 4.947.070            | 7,926         | 62.418.666            |
| 480       | BQMall                  | 1.167.580          | 2,818        | 25.595.973           | 61,774        | 128.483           | 0,310        | 11.868.262            | 28,643        | 2.674.600            | 6,455         | 41.434.898            |
| 480       | PartyScene              | 1.420.526          | 2,527        | 31.693.591           | 56,371        | 185.922           | 0,331        | 18.898.063            | 33,613        | 4.024.710            | 7,158         | 56.222.812            |
| 480       | RaceHorsesC             | 2.412.746          | 1,906        | 57.775.587           | 45,640        | 503.040           | 0,397        | 53.332.916            | 42,130        | 12.566.835           | 9,927         | 126.591.124           |
| 720       | FourPeople              | 2.156.679          | 3,351        | 47.122.240           | 73,224        | 314.038           | 0,488        | 11.272.831            | 17,517        | 3.488.162            | 5,420         | 64.353.950            |
| 720       | Johnny                  | 1.844.720          | 4,011        | 39.687.605           | 86,303        | 231.429           | 0,503        | 2.712.899             | 5,899         | 1.509.797            | 3,283         | 45.986.450            |
| 720       | KristenAndSara          | 2.142.067          | 3,278        | 47.741.941           | 73,061        | 388.935           | 0,595        | 11.277.040            | 17,258        | 3.795.174            | 5,808         | 65.345.156            |
| 720       | SlideEditing            | 475.422            | 2,283        | 10.966.449           | 52,651        | 52.263            | 0,251        | 7.715.561             | 37,043        | 1.618.712            | 7,772         | 20.828.406            |
| 720       | SlideShow               | 738.705            | 1,028        | 18.804.192           | 26,169        | 107.322           | 0,149        | 45.879.122            | 63,847        | 6.328.505            | 8,807         | 71.857.845            |
| 1080      | ArenaOfValor            | 5.547.475          | 1,911        | 122.554.481          | 42,214        | 804.619           | 0,277        | 136.167.991           | 46,903        | 25.241.507           | 8,694         | 290.316.073           |
| 1080      | BasketballDrive         | 5.416.632          | 1,441        | 134.220.475          | 35,719        | 1.263.852         | 0,336        | 193.453.300           | 51,482        | 41.413.558           | 11,021        | 375.767.817           |
| 1080      | BOTerrace               | 3.497.724          | 3,896        | 69.514.096           | 77,435        | 383.120           | 0,427        | 13.069.302            | 14,559        | 3.306.218            | 3,683         | 89.770.460            |
| 1080      | Cactus                  | 4.080.094          | 2,546        | 87.114.007           | 54,353        | 720.059           | 0,449        | 56.388.628            | 35,182        | 11.973.008           | 7,470         | 160.275.796           |
| 1080      | MarketPlace             | 20.080.134         | 0,862        | 527.343.076          | 22,636        | 319.644           | 0,014        | 1.494.912.503         | 64,168        | 287.032.273          | 12,321        | 2.329.687.630         |
| 4k        | Campfirebt709videoRange | 84.984.474         | 0,991        | 2.345.717.745        | 27,359        | 3.432.176         | 0,040        | 5.261.911.324         | 61,371        | 877.834.097          | 10,238        | 8.573.879.816         |
| 4k        | CatRobotjvet            | 17.859.821         | 2,116        | 404.816.410          | 47,967        | 3.956.493         | 0,469        | 345.311.115           | 40,916        | 72.010.460           | 8,533         | 843.954.299           |
| 4k        | DaylightRoad            | 23.087.245         | 2,246        | 525.568.987          | 51,123        | 5.997.405         | 0,583        | 390.337.286           | 37,969        | 83.049.911           | 8,078         | 1.028.040.834         |
| 4k        | FoodMarket              | 10.306.895         | 1,178        | 264.795.030          | 30,275        | 2.516.741         | 0,288        | 501.129.951           | 57,297        | 95.872.965           | 10,962        | 874.621.582           |
| 4k        | RitualDance             | 19.767.226         | 0,926        | 549.560.924          | 25,748        | 426.969           | 0,020        | 1.341.365.867         | 62,846        | 223.242.192          | 10,459        | 2.134.363.178         |
| 4k        | Tango                   | 22.835.171         | 1,478        | 564.054.201          | 36,508        | 5.841.770         | 0,378        | 797.097.302           | 51,591        | 155.189.120          | 10,044        | 1.545.017.564         |
| 4k        | VideoParkRunning        | 35.991.690         | 2,330        | 777.698.568          | 50,350        | 6.511.478         | 0,422        | 595.980.019           | 38,585        | 128.408.241          | 8,313         | 1.544.589.996         |
|           | <b>TOTAL</b>            | <b>267.340.283</b> | <b>1,314</b> | <b>6.687.283.684</b> | <b>32,869</b> | <b>34.321.389</b> | <b>0,169</b> | <b>11.310.851.885</b> | <b>55,594</b> | <b>2.045.527.113</b> | <b>10,054</b> | <b>20.345.324.353</b> |

### 5.2.2.1 Resolução 480p

A análise dos resultados, focada na codificação de vídeo com resolução de 480, traz à tona resultados significativos para avaliação. Eficiência, neste contexto, é definida como a capacidade do subtrator aproximado de reduzir o número de operações SAD executadas em comparação ao subtrator *baseline*. A Tabela 17 apresenta a variação do número de operações de SAD nas diversas etapas do TZS, considerando os subtratores aproximados usados na última seção. Na Tabela 17 estão apresentados os percentuais de variação dos resultados dos subtratores aproximados em relação ao subtrator *baseline* para as quatro sequências de vídeos de resolução 480p: *BQMall*, *PartyScene*, *RaceHorsesC* e *BasketballDrill*. Esses resultados proporcionam uma visão detalhada sobre o desempenho comparativo entre o subtrator aproximado e o subtrator *baseline*, enfatizando a capacidade de otimização de eficiência através da redução de operações SAD.

Os resultados indicam variações na eficácia, comparando o subtrator *baseline* com os subtratores aproximados utilizados no algoritmo TZS. Os vídeos *BQMall* e *PartyScene* apresentam eficiências superiores, com médias da coluna Total de -0,876 e -0,538, respectivamente, refletindo uma maior redução no número de operações SAD em comparação com o subtrator *baseline*. Em contraste, *BasketballDrill* e *RaceHorsesC* mostram eficiências menores, com médias da coluna Total de -0,054 e -0,262, respectivamente. Esta análise sugere que certos tipos de movimento ou padrões nos vídeos são mais susceptíveis à otimização por subtratores aproximados. As variações observadas na eficiência dos subtratores aproximados apontam para uma correlação entre o tipo de vídeo e a performance dos subtratores. Especificamente, vídeos como *BQMall* e *PartyScene* beneficiam-se significativamente de subtratores aproximados, sugerindo que esses subtratores podem ser mais adequados para conteúdos com características similares.

As diferentes etapas do algoritmo TZS mostram variações na eficiência dos subtratores aproximados. A etapa de Busca Refinada registra a maior redução média no número de operações SAD, com uma média de -1,218, enquanto as etapas de Varredura e Busca Inicial exibem as menores reduções, com médias de -0,311 e -0,337, respectivamente. Isso indica que melhorias nos subtratores utilizados nestas fases menos eficientes poderiam potencializar a performance geral do algoritmo. A eficiência notória na fase de Busca Refinada (Ref) sugere que subtratores aproximados são particularmente eficazes quando a tolerância à imprecisão é uma opção viável. Por outro lado, as fases de Varredura e Busca Inicial requerem uma abordagem mais precisa, indicando a necessidade de ajustes nos subtratores para otimizar essas etapas.

A comparação entre os subtratores apresenta que  $AXCS_1$  e  $AppS_4$  são os mais eficientes, com médias de redução no número de operações SAD de -1,085 e -1,005, respectivamente. Em contraste, o subtrator  $AXCS_3$  mostra um aumento nas opera-

ções, com uma média de 0,378, destacando-se como menos eficiente. Os resultados apontam que o  $AXCS_1$  e  $AppS_4$  como os subtratores mais adequados para situações que exigem alta eficiência. A análise também evidencia a importância de evitar o uso de  $AXCS_3$  em cenários que demandam redução significativa no número de operações *SAD*.

Tabela 17 – Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações.Resolução 480p

| Subtrator                                    | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
|----------------------------------------------|--------|--------|--------|--------------|--------|----------|
| <i>AppS</i> <sub>1</sub>                     | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,00     |
| <i>AppS</i> <sub>2</sub>                     | -0,080 | 0,052  | 0,994  | 1,464        | 0,022  | 0,54     |
| <i>AppS</i> <sub>3</sub>                     | -0,318 | -0,363 | 0,743  | 0,112        | -0,352 | -0,20    |
| <i>AppS</i> <sub>4</sub>                     | 0,128  | 0,151  | 0,222  | -0,316       | 0,082  | -0,02    |
| <i>AXCS</i> <sub>1</sub>                     | -0,570 | -0,538 | 0,020  | -0,244       | -1,662 | -0,54    |
| <i>AXCS</i> <sub>1</sub> <sub>2</sub>        | -1,359 | -1,361 | -0,230 | -0,137       | -0,463 | -0,89    |
| <i>AXCS</i> <sub>1</sub> <sub>3</sub>        | -0,501 | -0,178 | -0,284 | 0,406        | -1,582 | -0,11    |
| <i>AXCS</i> <sub>1</sub> <sub>4</sub>        | 0,485  | 0,525  | 0,343  | -0,232       | -0,755 | 0,16     |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub>        | 0,347  | 0,745  | 1,066  | -0,608       | -2,289 | 0,02     |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub>        | -0,492 | -0,126 | 0,740  | -0,414       | -0,298 | -0,26    |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub>        | 0,556  | 0,829  | 0,611  | 0,249        | -1,797 | 0,41     |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub>        | -0,033 | 0,290  | 0,508  | -0,402       | -2,191 | -0,17    |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub>        | -1,084 | -0,393 | -0,476 | -1,894       | -5,599 | -1,40    |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub>        | 0,450  | 0,839  | 1,704  | 0,250        | -0,219 | 0,55     |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub>        | 0,450  | 1,458  | 5,018  | -0,255       | -3,271 | 0,46     |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub>        | 0,411  | 1,127  | 2,518  | 0,396        | -2,272 | 0,58     |
| Video:BasketballDrill TI: 6,913 SI: 5970,540 |        |        |        |              |        |          |
| Subtrator                                    | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                     | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,00     |
| <i>AppS</i> <sub>2</sub>                     | 0,762  | 0,581  | -0,089 | -0,374       | -0,786 | 0,19     |
| <i>AppS</i> <sub>3</sub>                     | -0,512 | -0,431 | -0,222 | 0,094        | -0,343 | -0,27    |
| <i>AppS</i> <sub>4</sub>                     | -0,768 | -0,966 | -1,580 | -2,134       | -2,068 | -1,45    |
| <i>AXCS</i> <sub>1</sub> <sub>1</sub>        | -0,032 | -0,005 | -0,518 | 0,666        | -0,688 | 0,17     |
| <i>AXCS</i> <sub>1</sub> <sub>2</sub>        | -0,675 | -0,628 | -1,635 | 1,165        | 0,191  | 0,00     |
| <i>AXCS</i> <sub>1</sub> <sub>3</sub>        | 0,232  | 0,118  | -0,805 | 0,170        | -0,089 | 0,13     |
| <i>AXCS</i> <sub>1</sub> <sub>4</sub>        | 0,703  | 0,548  | -0,882 | -1,074       | -0,307 | -0,03    |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub>        | 0,899  | 1,007  | -0,714 | 0,540        | 0,721  | 0,85     |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub>        | -0,015 | -0,208 | -1,118 | 0,151        | -0,210 | -0,09    |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub>        | -0,484 | -0,330 | -0,647 | -0,942       | -0,727 | -0,58    |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub>        | 0,411  | 0,596  | -0,508 | -0,604       | -0,658 | 0,11     |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub>        | -0,848 | -0,590 | -2,011 | -2,388       | -2,383 | -1,34    |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub>        | -0,987 | -1,100 | -1,799 | -2,420       | -1,885 | -1,62    |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub>        | 0,041  | 0,406  | -0,567 | -1,576       | -1,761 | -0,41    |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub>        | 1,296  | 1,286  | -0,733 | -1,768       | -0,922 | 0,15     |
| Video:PartyScene. TI: 7,271 SI: 14718,577    |        |        |        |              |        |          |
| Subtrator                                    | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                     | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,00     |
| <i>AppS</i> <sub>2</sub>                     | -0,526 | -0,506 | -0,898 | -0,191       | 0,073  | -0,33    |
| <i>AppS</i> <sub>3</sub>                     | -1,061 | -0,946 | -0,885 | 0,196        | -0,033 | -0,41    |
| <i>AppS</i> <sub>4</sub>                     | -0,986 | -0,920 | -2,099 | -0,034       | -0,799 | -0,57    |
| <i>AXCS</i> <sub>1</sub> <sub>1</sub>        | -1,066 | -1,055 | -2,243 | -0,791       | -1,669 | -1,03    |
| <i>AXCS</i> <sub>1</sub> <sub>2</sub>        | 0,611  | 0,687  | 0,904  | 0,807        | 0,786  | 0,76     |
| <i>AXCS</i> <sub>1</sub> <sub>3</sub>        | -0,814 | -0,529 | -2,978 | -0,732       | -1,503 | -0,74    |
| <i>AXCS</i> <sub>1</sub> <sub>4</sub>        | -0,164 | 0,014  | -2,307 | 0,189        | -0,820 | -0,01    |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub>        | -0,831 | -0,602 | -5,910 | -1,224       | -3,069 | -1,14    |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub>        | -1,826 | -1,706 | -3,142 | -0,594       | -1,101 | -1,23    |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub>        | -1,132 | -0,792 | -4,894 | -0,564       | -2,736 | -0,93    |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub>        | -0,455 | -0,038 | -4,269 | -0,711       | -2,611 | -0,60    |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub>        | -0,363 | 0,189  | -6,853 | -1,567       | -4,702 | -1,06    |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub>        | -0,499 | -0,055 | -3,004 | -0,318       | -1,120 | -0,30    |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub>        | -0,123 | 0,740  | -6,554 | -0,960       | -3,422 | -0,42    |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub>        | -0,827 | -0,060 | -7,821 | -1,157       | -3,547 | -0,59    |
| Video:RaceHorseC. TI.: 21,905 SI.:8987,960   |        |        |        |              |        |          |
| Subtrator                                    | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                     | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,00     |
| <i>AppS</i> <sub>2</sub>                     | -0,526 | -0,506 | -0,898 | -0,191       | 0,073  | -0,33    |
| <i>AppS</i> <sub>3</sub>                     | -1,061 | -0,946 | -0,885 | 0,196        | -0,033 | -0,41    |
| <i>AppS</i> <sub>4</sub>                     | -0,986 | -0,920 | -2,099 | -0,034       | -0,799 | -0,57    |
| <i>AXCS</i> <sub>1</sub> <sub>1</sub>        | -1,066 | -1,055 | -2,243 | -0,791       | -1,669 | -1,03    |
| <i>AXCS</i> <sub>1</sub> <sub>2</sub>        | 0,611  | 0,687  | 0,904  | 0,807        | 0,786  | 0,76     |
| <i>AXCS</i> <sub>1</sub> <sub>3</sub>        | -0,814 | -0,529 | -2,978 | -0,732       | -1,503 | -0,74    |
| <i>AXCS</i> <sub>1</sub> <sub>4</sub>        | -0,164 | 0,014  | -2,307 | 0,189        | -0,820 | -0,01    |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub>        | -0,831 | -0,602 | -5,910 | -1,224       | -3,069 | -1,14    |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub>        | -1,826 | -1,706 | -3,142 | -0,594       | -1,101 | -1,23    |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub>        | -1,132 | -0,792 | -4,894 | -0,564       | -2,736 | -0,93    |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub>        | -0,455 | -0,038 | -4,269 | -0,711       | -2,611 | -0,60    |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub>        | -0,363 | 0,189  | -6,853 | -1,567       | -4,702 | -1,06    |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub>        | -0,499 | -0,055 | -3,004 | -0,318       | -1,120 | -0,30    |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub>        | -0,123 | 0,740  | -6,554 | -0,960       | -3,422 | -0,42    |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub>        | -0,827 | -0,060 | -7,821 | -1,157       | -3,547 | -0,59    |

### 5.2.2.2 Resolução 720p

A análise realizada focando na resolução 720 tem seus resultados detalhados nas Tabelas 18 e 19.

Ao analisarmos as tendências de eficiência do algoritmo *Test Zone Search* (TZS) com relação ao número de operações SAD executadas em diferentes vídeos, observamos variações significativas que refletem as particularidades de cada vídeo. Por exemplo, o vídeo *FourPeople* exibe um aumento moderado nas operações de SAD em todas as fases, alcançando uma média da coluna Total de 5,67. Em contraste, o vídeo *Johnny* mostra valores significativamente mais altos, especialmente na fase de varredura, onde a média atinge 121,56. Essa fase resulta em um total médio de 18,99, indicando uma baixa eficiência devido às exigências específicas de precisão deste vídeo. Vale destacar que, neste contexto, ‘*eficiência*’ é definida como a capacidade do subtrator de reduzir o número de operações SAD executadas.

O vídeo *KristenAndSara* destaca-se por uma alta eficiência, com valores médios baixos em quase todas as fases, exceto na fase de Busca Inicial (BI) e Dois Pontos (DP), alcançando um valor total médio de apenas 0,79. De maneira interessante, *SlideEdit* mostra uma média total negativa de -0,65, sugerindo uma redução nas operações SAD e indicando eficiência. Apesar de valores mais altos nas fases de varredura e Busca Refinada (Ref), a média negativa em DP ajuda a equilibrar esses aumentos. Similar ao vídeo *FourPeople*, *SlideShow* apresenta valores moderadamente altos em todas as fases, com um total médio de 4,07, refletindo uma eficiência moderada.

Existem variações consideráveis nas tendências de SAD entre os vídeos, o que pode ser atribuído às características visuais e de movimento específicas de cada um. *Johnny* e *SlideEdit* mostram extremos em eficiência, com *Johnny* necessitando de muitas operações SAD, enquanto *SlideEdit* exibe uma notável eficiência com reduções no número de SADs executados. Vídeos como *KristenAndSara* também demonstram boa eficiência, enquanto *FourPeople* e *SlideShow* apresentam um desempenho moderado. A escolha do subtrator influencia significativamente a eficiência do algoritmo em diferentes etapas, com etapas como Varredura e Busca Refinada (Ref) sendo particularmente sensíveis à escolha do subtrator.

O subtrator  $AXCS1_2$  se destaca como o mais eficiente, aparecendo duas vezes como o melhor em redução no número de operações SAD, o que demonstra sua capacidade de otimizar significativamente o processamento. Outros subtratores, como  $AXCS3_1$ ,  $AXCS3_4$ ,  $AppS_2$  e  $AppS_4$ , também mostram bons resultados, cada um contribuindo com uma ocorrência onde houve uma significativa redução das operações SAD.

Em contraste, o subtrator  $AXCS2_4$  é frequentemente identificado como o menos eficiente, liderando três vezes como o subtrator com pior desempenho em diferentes etapas do algoritmo, indicando uma tendência de aumento no número de operações

SAD quando utilizado. Os subtratores  $AXCS1_4$  e  $AXCS3_2$  também aparecem cada um uma vez como o menos eficiente, sugerindo que em certas etapas eles podem não ser tão eficazes.

Estas tendências têm implicações práticas importantes para a seleção de subtratores em sistemas de codificação de vídeo. Subtratores como  $AXCS1_2$  e  $AXCS3_1$  são recomendados para uso em cenários onde a eficiência é uma prioridade, oferecendo reduções significativas nas operações SAD. Por outro lado, subtratores como  $AXCS2_4$  devem ser evitados em aplicações onde a redução de operações é crítica, devido ao seu desempenho frequentemente inferior.



Tabela 18 – Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução:720

| Subtrator                             | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
|---------------------------------------|--------|--------|--------|--------------|--------|----------|
| <i>AppS</i> <sub>1</sub>              | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0        |
| <i>AppS</i> <sub>2</sub>              | 0,241  | 0,503  | 1,953  | 1,607        | 0,938  | 0,232    |
| <i>AppS</i> <sub>3</sub>              | 0,641  | 1,324  | 2,571  | 3,632        | 3,231  | 1,033    |
| <i>AppS</i> <sub>4</sub>              | 0,981  | 2,850  | 3,542  | 25,338       | 16,434 | 1,842    |
| <i>AXCS</i> <sub>1</sub>              | -0,461 | -0,478 | -0,332 | -0,714       | -0,775 | 2,614    |
| <i>AXCS</i> <sub>2</sub>              | 0,994  | 1,035  | 0,673  | 3,349        | 1,497  | 3,468    |
| <i>AXCS</i> <sub>3</sub>              | 0,284  | 0,690  | 2,601  | 3,411        | 3,214  | 4,276    |
| <i>AXCS</i> <sub>4</sub>              | 0,647  | 1,919  | 5,114  | 26,007       | 17,253 | 5,121    |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub> | 1,036  | 1,057  | 1,074  | 2,779        | 1,767  | 5,975    |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub> | -0,339 | 0,096  | 1,956  | 2,675        | 2,194  | 6,794    |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub> | 0,109  | 1,008  | 3,169  | 8,887        | 6,876  | 7,668    |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub> | 0,418  | 2,400  | 4,189  | 36,082       | 21,590 | 8,548    |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub> | 0,105  | 0,121  | 0,163  | 0,138        | -0,713 | 9,422    |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub> | 0,826  | 1,063  | 2,264  | 3,673        | 1,943  | 10,333   |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub> | 0,248  | 0,799  | 2,447  | 2,635        | 2,677  | 11,220   |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub> | -0,497 | 0,807  | 1,677  | 11,831       | 10,508 | 12,112   |

Video:FourPeople.TI:2,220 e SI:8987,960

| Subtrator                             | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
|---------------------------------------|--------|--------|--------|--------------|--------|----------|
| <i>AppS</i> <sub>1</sub>              | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>              | -0,579 | 0,057  | 0,932  | -0,580       | 0,352  | -0,110   |
| <i>AppS</i> <sub>3</sub>              | 0,632  | 1,809  | 0,207  | 1,757        | 2,197  | -0,163   |
| <i>AppS</i> <sub>4</sub>              | -0,222 | 2,697  | -3,317 | 52,300       | 26,008 | -0,279   |
| <i>AXCS</i> <sub>1</sub>              | 0,885  | 0,819  | 0,760  | -0,151       | 0,394  | -0,334   |
| <i>AXCS</i> <sub>2</sub>              | -0,029 | -0,087 | -0,393 | 0,735        | -0,528 | -0,449   |
| <i>AXCS</i> <sub>3</sub>              | 1,151  | 1,695  | 1,556  | 1,180        | 4,106  | -0,500   |
| <i>AXCS</i> <sub>4</sub>              | -1,759 | 0,094  | -1,140 | 41,166       | 21,735 | -0,673   |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub> | 0,154  | 0,130  | 0,418  | 1,219        | 0,467  | -0,700   |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub> | -0,137 | 0,837  | 0,313  | 2,970        | 4,192  | -0,792   |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub> | 0,612  | 2,510  | 0,327  | 20,784       | 12,560 | -0,853   |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub> | -0,038 | 3,577  | -2,111 | 99,289       | 44,499 | -0,953   |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub> | -0,191 | -0,039 | -0,310 | -0,755       | -0,427 | -1,038   |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub> | -0,160 | 0,407  | 1,255  | -0,183       | 1,546  | -1,117   |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub> | -1,087 | -0,024 | -1,120 | 2,441        | 0,544  | -1,222   |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub> | 1,369  | 3,763  | -1,286 | 29,811       | 17,672 | -1,228   |

Video:SlideEditing.TI: 10,201 e SI:1592,399

Video:Johnny.TI: 2,295 e SI:3942,258

| Subtrator                             | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%)  | Total(%) |
|---------------------------------------|--------|--------|--------|--------------|---------|----------|
| <i>AppS</i> <sub>1</sub>              | 0,000  | 0,000  | 0,000  | 0,000        | 0,000   | 0,000    |
| <i>AppS</i> <sub>2</sub>              | -0,076 | 0,606  | 3,831  | 11,253       | 6,693   | 1,695    |
| <i>AppS</i> <sub>3</sub>              | 0,352  | 2,103  | 6,730  | 59,167       | 24,146  | 3,534    |
| <i>AppS</i> <sub>4</sub>              | -0,045 | 3,435  | 4,064  | 354,765      | 84,927  | 5,476    |
| <i>AXCS</i> <sub>1</sub>              | -0,421 | -0,432 | -0,650 | 0,799        | -0,632  | 7,570    |
| <i>AXCS</i> <sub>2</sub>              | 1,130  | 1,183  | 1,265  | 3,302        | 1,789   | 9,908    |
| <i>AXCS</i> <sub>3</sub>              | -0,054 | 1,215  | 4,614  | 45,395       | 23,295  | 12,318   |
| <i>AXCS</i> <sub>4</sub>              | 0,298  | 3,309  | 8,715  | 315,590      | 93,867  | 14,994   |
| <i>AXCS</i> <sub>2</sub> <sub>1</sub> | 0,830  | 0,823  | 0,231  | 3,014        | 1,761   | 17,901   |
| <i>AXCS</i> <sub>2</sub> <sub>2</sub> | 0,088  | 1,540  | 6,910  | 44,110       | 23,803  | 21,003   |
| <i>AXCS</i> <sub>2</sub> <sub>3</sub> | 0,452  | 2,763  | 3,613  | 181,823      | 55,658  | 24,424   |
| <i>AXCS</i> <sub>2</sub> <sub>4</sub> | 1,157  | 5,190  | 2,921  | 614,056      | 144,806 | 28,161   |
| <i>AXCS</i> <sub>3</sub> <sub>1</sub> | 0,110  | 0,653  | 2,095  | 8,858        | 6,722   | 32,162   |
| <i>AXCS</i> <sub>3</sub> <sub>2</sub> | -0,995 | -0,231 | 2,147  | 12,471       | 6,397   | 36,530   |
| <i>AXCS</i> <sub>3</sub> <sub>3</sub> | 0,921  | 2,363  | 7,767  | 43,107       | 21,193  | 41,432   |
| <i>AXCS</i> <sub>3</sub> <sub>4</sub> | 1,306  | 4,079  | 6,509  | 247,280      | 75,907  | 46,725   |

Video:SlideShow.TI: 32,545 e SI:9027,841

Tabela 19 – Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo KristenAndSara, TI:2,148 , SI: 7247,037.

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada (%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|--------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000        | 0,000    |
| <i>AppS</i> <sub>2</sub>  | -0,966              | -0,941           | -2,349         | -1,231       | -0,479       | -1,104   |
| <i>AppS</i> <sub>3</sub>  | 0,978               | 1,538            | 0,105          | 1,387        | 1,448        | 1,433    |
| <i>AppS</i> <sub>4</sub>  | 0,703               | 1,185            | -0,072         | 0,655        | 0,466        | 0,773    |
| <i>AXCS</i> <sub>11</sub> | -0,199              | 0,589            | 1,614          | -0,438       | -0,231       | -0,168   |
| <i>AXCS</i> <sub>12</sub> | -0,037              | 0,735            | -2,069         | 0,655        | 0,474        | 0,648    |
| <i>AXCS</i> <sub>13</sub> | -0,292              | 0,601            | 1,707          | 1,090        | 0,370        | 0,894    |
| <i>AXCS</i> <sub>14</sub> | -0,098              | 0,887            | 1,622          | 0,856        | 0,176        | 0,796    |
| <i>AXCS</i> <sub>21</sub> | 0,299               | 2,999            | 5,920          | 1,192        | 0,717        | 1,590    |
| <i>AXCS</i> <sub>22</sub> | -0,593              | 0,390            | 1,079          | 0,872        | 0,486        | 0,702    |
| <i>AXCS</i> <sub>23</sub> | -0,928              | 1,727            | 6,193          | 2,408        | 2,205        | 2,190    |
| <i>AXCS</i> <sub>24</sub> | 0,761               | 2,348            | 2,441          | 2,100        | 0,249        | 1,995    |
| <i>AXCS</i> <sub>31</sub> | -1,860              | 1,124            | 3,721          | -0,220       | -0,727       | 0,034    |
| <i>AXCS</i> <sub>32</sub> | -1,157              | 0,714            | 3,251          | -0,478       | 0,669        | -0,102   |
| <i>AXCS</i> <sub>33</sub> | -1,025              | 2,468            | 3,500          | 0,698        | 1,992        | 1,216    |
| <i>AXCS</i> <sub>34</sub> | 0,914               | 3,699            | 7,366          | 1,152        | 1,445        | 1,810    |

### 5.2.2.3 Resolução 1080p

Os resultados da análise para vídeos de 1080p estão nas Tabelas 20 e 21, de forma similar às análises apresentadas nas seções anteriores.

Arena apresenta valores moderados em todas as fases, indicando um equilíbrio geral, com nenhuma característica específica dominando em termos de número de SADs executados. BQTerrace se destaca por valores excepcionalmente altos na fase de Dois Ponto (DP) e um total geral significativamente maior do que outros vídeos, o que sugere que as características deste vídeo podem demandar mais operações SAD ou que os subtratores usados não foram tão eficazes.

BasketBallDrive mostra uma tendência geral de redução nas operações SAD, especialmente nas etapas de varredura e Busca Refinada (Ref), indicando que este vídeo possui características que favorecem a eficiência dos subtratores em minimizar essas operações. Cactus, por sua vez, mostra uma pequena redução em DP e Busca Refinada (Ref), com um total geral levemente positivo, refletindo uma eficiência moderada dos subtratores utilizados.

*MarketPlace* é notável por apresentar a única média redução no total entre todos os vídeos analisados, sugerindo um bom desempenho em reduzir operações SAD em todas as fases. Em contraste, *RitualDance* tem aumentos no número de SADs executados, particularmente nas fases de Decisão de Predição (DP) e Varredura, com um total significativo. Esse aumento pode indicar desafios particulares que resultam em uma maior necessidade de operações SAD, ou sugerir que os subtratores utilizados podem não ser tão eficientes para este tipo de conteúdo.

Os subtratores da família AXCS1, como *AXCS*<sub>11</sub> a *AXCS*<sub>14</sub>, mostram uma variação moderada nas operações SAD, com *AXCS*<sub>14</sub> exibindo a maior média na redução, indicando uma eficácia geral em reduzir operações. Por outro lado, *AXCS*<sub>21</sub> a

$AXCS2_4$  e  $AXCS3_1$  a  $AXCS3_4$  apresentam aumentos nas operações SAD, especialmente notáveis em Dois Pontos (DP), Varredura e Busca Refinada (Ref), com  $AXCS3_3$  e  $AXCS3_2$  destacando-se por aumentos consideráveis, sugerindo uma ineficácia desses subtratores que pode demandar reavaliações ou ajustes.

Os subtratores AppS, de  $AppS_1$  a  $AppS_4$ , demonstram uma tendência geral de redução nas operações SAD, com  $AppS_2$  e  $AppS_3$  exibindo reduções em quase todas as fases, o que sugere uma eficiência superior.  $AppS_1$ , apresentando desempenho neutro em todas as fases, pode ser útil para manter a consistência do algoritmo.

Conclui-se que a escolha do subtrator tem um impacto substancial sobre a eficiência do algoritmo Test Zone Search (TZS), com subtratores como  $AppS_2$  e  $AppS_3$  oferecendo um desempenho robusto na redução de operações SAD, enquanto outros, como  $AXCS3_3$  e  $AXCS3_2$ , podem necessitar de ajustes ou substituições para melhorar a eficiência em contextos práticos. A análise das médias das operações SAD para cada etapa do TZS revela que as etapas DP e Busca Refinada são críticas para focar em melhorias, enquanto PM mostra ser a menos impactada. Ajustar a escolha e configuração dos subtratores com base nas demandas específicas de cada fase pode levar a uma redução significativa no número total de operações SAD necessárias.

Tabela 20 – Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Resolução 1080p.

| Subtrator                                  | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
|--------------------------------------------|--------|--------|---------|--------------|--------|----------|
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | -0,156 | -0,075 | 0,188   | 0,063        | 0,298  | 0,077    |
| <i>AppS</i> <sub>3</sub>                   | 0,543  | 0,561  | 0,135   | 0,668        | 0,339  | 0,352    |
| <i>AppS</i> <sub>4</sub>                   | 0,522  | 0,540  | 0,060   | 0,683        | -0,193 | 0,261    |
| <i>AXCS</i> <sub>11</sub>                  | 0,401  | 0,538  | -0,319  | 0,720        | -0,603 | 0,042    |
| <i>AXCS</i> <sub>12</sub>                  | 0,883  | 0,942  | 0,247   | 1,001        | 0,657  | 0,607    |
| <i>AXCS</i> <sub>13</sub>                  | 0,034  | 0,258  | -0,247  | 0,792        | -0,528 | -0,049   |
| <i>AXCS</i> <sub>14</sub>                  | 0,589  | 0,676  | 0,318   | 1,204        | -0,474 | 0,419    |
| <i>AXCS</i> <sub>21</sub>                  | 0,741  | 1,460  | -0,485  | 2,210        | -1,505 | 0,293    |
| <i>AXCS</i> <sub>22</sub>                  | 0,669  | 0,878  | -0,080  | 1,925        | -0,056 | 0,359    |
| <i>AXCS</i> <sub>23</sub>                  | 0,435  | 1,133  | -0,869  | 2,451        | -1,624 | -0,047   |
| <i>AXCS</i> <sub>24</sub>                  | 0,687  | 1,100  | -0,519  | 1,250        | -1,476 | 0,122    |
| <i>AXCS</i> <sub>31</sub>                  | 0,308  | 1,871  | 0,106   | 5,163        | -1,014 | 0,777    |
| <i>AXCS</i> <sub>32</sub>                  | 0,822  | 1,665  | 1,612   | 5,226        | 1,338  | 1,621    |
| <i>AXCS</i> <sub>33</sub>                  | 0,538  | 1,828  | 0,807   | 5,980        | -0,615 | 1,134    |
| <i>AXCS</i> <sub>34</sub>                  | 0,719  | 1,760  | -0,655  | 3,983        | -1,073 | 0,381    |
| Video:ArenaOfValor. TI: 7,645 SI: 7518,191 |        |        |         |              |        |          |
| Subtrator                                  | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | -0,810 | -0,831 | -1,264  | -0,604       | -0,583 | -0,914   |
| <i>AppS</i> <sub>3</sub>                   | -0,671 | -0,635 | -1,861  | -0,180       | -0,940 | -0,850   |
| <i>AppS</i> <sub>4</sub>                   | 0,911  | 0,878  | 1,531   | 1,272        | -0,025 | 0,979    |
| <i>AXCS</i> <sub>11</sub>                  | 0,570  | 0,737  | 3,012   | 0,141        | 1,394  | 1,106    |
| <i>AXCS</i> <sub>12</sub>                  | 0,478  | 0,558  | -0,057  | 0,464        | 1,268  | 0,510    |
| <i>AXCS</i> <sub>13</sub>                  | -0,164 | 0,134  | 4,036   | 1,032        | 2,508  | 0,775    |
| <i>AXCS</i> <sub>14</sub>                  | 0,192  | 0,277  | 0,794   | 0,478        | -0,156 | 0,341    |
| <i>AXCS</i> <sub>21</sub>                  | 0,562  | 1,197  | 10,299  | 1,274        | 8,114  | 2,774    |
| <i>AXCS</i> <sub>22</sub>                  | 0,933  | 1,210  | 6,527   | 0,882        | 4,378  | 2,125    |
| <i>AXCS</i> <sub>23</sub>                  | 0,502  | 1,158  | 25,750  | 1,321        | 13,653 | 5,193    |
| <i>AXCS</i> <sub>24</sub>                  | 0,243  | 0,775  | 7,475   | 1,861        | 5,321  | 1,911    |
| <i>AXCS</i> <sub>31</sub>                  | 0,535  | 1,720  | 42,144  | 0,435        | 27,095 | 8,509    |
| <i>AXCS</i> <sub>32</sub>                  | 0,724  | 1,764  | 27,820  | 3,100        | 17,328 | 6,124    |
| <i>AXCS</i> <sub>33</sub>                  | 1,422  | 3,191  | 107,942 | 1,389        | 50,331 | 20,157   |
| <i>AXCS</i> <sub>34</sub>                  | 1,396  | 2,590  | 32,147  | 1,535        | 21,182 | 7,582    |
| Video:BQTerrace. TI: 6,056 SI: 15470,666   |        |        |         |              |        |          |
| Subtrator                                  | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | -0,232 | 0,231  | -0,550  | 0,215        | -0,265 | -0,075   |
| <i>AppS</i> <sub>3</sub>                   | 0,545  | 0,421  | -0,347  | 0,165        | -0,332 | 0,111    |
| <i>AppS</i> <sub>4</sub>                   | -0,122 | -0,273 | -1,149  | -0,611       | -1,549 | -0,677   |
| <i>AXCS</i> <sub>11</sub>                  | -0,202 | 0,127  | -0,487  | 1,261        | -0,577 | -0,150   |
| <i>AXCS</i> <sub>12</sub>                  | -0,264 | -0,227 | 0,040   | 0,171        | 0,022  | -0,120   |
| <i>AXCS</i> <sub>13</sub>                  | 0,044  | 0,197  | -0,933  | 1,394        | -1,449 | -0,321   |
| <i>AXCS</i> <sub>14</sub>                  | -0,836 | -0,843 | -1,102  | -0,810       | -1,751 | -1,023   |
| <i>AXCS</i> <sub>21</sub>                  | 0,331  | 0,945  | -0,941  | 1,474        | -1,551 | 0,090    |
| <i>AXCS</i> <sub>22</sub>                  | 0,384  | 0,633  | -0,267  | 1,424        | 0,008  | 0,277    |
| <i>AXCS</i> <sub>23</sub>                  | -0,414 | 0,092  | -1,170  | 0,867        | -1,885 | -0,520   |
| <i>AXCS</i> <sub>24</sub>                  | 0,541  | 0,480  | -0,980  | 0,970        | -1,782 | -0,185   |
| <i>AXCS</i> <sub>31</sub>                  | -0,078 | 1,217  | -1,467  | 1,627        | -1,440 | 0,041    |
| <i>AXCS</i> <sub>32</sub>                  | 0,048  | 1,028  | 2,853   | 3,713        | 2,196  | 1,746    |
| <i>AXCS</i> <sub>33</sub>                  | 0,561  | 2,079  | 2,300   | 2,885        | 0,934  | 2,051    |
| <i>AXCS</i> <sub>34</sub>                  | 0,981  | 2,240  | 2,870   | 2,987        | 0,979  | 2,364    |
| Video:Cactus.SI: 6,886 TI: 5411,983        |        |        |         |              |        |          |
| Subtrator                                  | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | 0,232  | 0,231  | -0,550  | 0,215        | -0,265 | -0,075   |
| <i>AppS</i> <sub>3</sub>                   | 0,545  | 0,421  | -0,347  | 0,165        | -0,332 | 0,111    |
| <i>AppS</i> <sub>4</sub>                   | -0,122 | -0,273 | -1,149  | -0,611       | -1,549 | -0,677   |
| <i>AXCS</i> <sub>11</sub>                  | -0,202 | 0,127  | -0,487  | 1,261        | -0,577 | -0,150   |
| <i>AXCS</i> <sub>12</sub>                  | -0,264 | -0,227 | 0,040   | 0,171        | 0,022  | -0,120   |
| <i>AXCS</i> <sub>13</sub>                  | 0,044  | 0,197  | -0,933  | 1,394        | -1,449 | -0,321   |
| <i>AXCS</i> <sub>14</sub>                  | -0,836 | -0,843 | -1,102  | -0,810       | -1,751 | -1,023   |
| <i>AXCS</i> <sub>21</sub>                  | 0,331  | 0,945  | -0,941  | 1,474        | -1,551 | 0,090    |
| <i>AXCS</i> <sub>22</sub>                  | 0,384  | 0,633  | -0,267  | 1,424        | 0,008  | 0,277    |
| <i>AXCS</i> <sub>23</sub>                  | -0,414 | 0,092  | -1,170  | 0,867        | -1,885 | -0,520   |
| <i>AXCS</i> <sub>24</sub>                  | 0,541  | 0,480  | -0,980  | 0,970        | -1,782 | -0,185   |
| <i>AXCS</i> <sub>31</sub>                  | -0,078 | 1,217  | -1,467  | 1,627        | -1,440 | 0,041    |
| <i>AXCS</i> <sub>32</sub>                  | 0,048  | 1,028  | 2,853   | 3,713        | 2,196  | 1,746    |
| <i>AXCS</i> <sub>33</sub>                  | 0,561  | 2,079  | 2,300   | 2,885        | 0,934  | 2,051    |
| <i>AXCS</i> <sub>34</sub>                  | 0,981  | 2,240  | 2,870   | 2,987        | 0,979  | 2,364    |

Tabela 21 – Comparação da Quantidade de SADs Executadas para Blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações.

| Subtrator                                  | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
|--------------------------------------------|--------|--------|--------|--------------|--------|----------|
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | -0,445 | -0,429 | -0,348 | -0,131       | -0,106 | -0,202   |
| <i>AppS</i> <sub>3</sub>                   | -0,456 | -0,376 | -0,085 | -0,085       | 0,020  | -0,145   |
| <i>AppS</i> <sub>4</sub>                   | -0,604 | -0,571 | -0,655 | -0,148       | -0,181 | -0,257   |
| <i>AXCS</i> <sub>11</sub>                  | 0,001  | -0,103 | -0,122 | -0,005       | 0,036  | -0,022   |
| <i>AXCS</i> <sub>12</sub>                  | -0,117 | -0,074 | -0,653 | 0,318        | 0,293  | 0,221    |
| <i>AXCS</i> <sub>13</sub>                  | -0,109 | -0,133 | -0,522 | 0,005        | -0,067 | -0,037   |
| <i>AXCS</i> <sub>14</sub>                  | -0,363 | -0,357 | -0,699 | -0,136       | -0,107 | -0,188   |
| <i>AXCS</i> <sub>21</sub>                  | -0,402 | -0,429 | -0,806 | -0,093       | -0,084 | -0,174   |
| <i>AXCS</i> <sub>22</sub>                  | -0,214 | -0,107 | -0,133 | 0,032        | 0,054  | -0,001   |
| <i>AXCS</i> <sub>23</sub>                  | -0,508 | -0,479 | -0,671 | -0,295       | -0,146 | -0,325   |
| <i>AXCS</i> <sub>24</sub>                  | -0,356 | -0,348 | 0,021  | -0,088       | -0,137 | -0,158   |
| <i>AXCS</i> <sub>31</sub>                  | -0,742 | -0,738 | -0,149 | -0,137       | -0,033 | -0,272   |
| <i>AXCS</i> <sub>32</sub>                  | -0,086 | -0,140 | 0,620  | 0,026        | 0,118  | -0,002   |
| <i>AXCS</i> <sub>33</sub>                  | -0,128 | -0,202 | -0,148 | -0,457       | -0,278 | -0,375   |
| <i>AXCS</i> <sub>34</sub>                  | 0,031  | -0,074 | -0,201 | -0,399       | -0,408 | -0,323   |
| Video:MarketPlace.TI: 28,965 e SI:5977,375 |        |        |        |              |        |          |
| Subtrator                                  | PM(%)  | BI(%)  | DP(%)  | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS</i> <sub>1</sub>                   | 0,000  | 0,000  | 0,000  | 0,000        | 0,000  | 0,000    |
| <i>AppS</i> <sub>2</sub>                   | 0,348  | 0,477  | 0,926  | 0,424        | 0,337  | 0,431    |
| <i>AppS</i> <sub>3</sub>                   | -0,504 | -0,417 | 0,712  | 0,226        | 0,246  | 0,051    |
| <i>AppS</i> <sub>4</sub>                   | 0,416  | 0,395  | -0,319 | 0,748        | -0,028 | 0,577    |
| <i>AXCS</i> <sub>11</sub>                  | -0,264 | 0,678  | 2,939  | 0,692        | 0,519  | 0,659    |
| <i>AXCS</i> <sub>12</sub>                  | 0,009  | 0,271  | 0,628  | 0,681        | 0,462  | 0,546    |
| <i>AXCS</i> <sub>13</sub>                  | -0,013 | 0,617  | 4,712  | 1,034        | 0,746  | 0,887    |
| <i>AXCS</i> <sub>14</sub>                  | 0,233  | 0,487  | 4,130  | 0,360        | -0,061 | 0,350    |
| <i>AXCS</i> <sub>21</sub>                  | 0,245  | 0,802  | 5,659  | 0,840        | 1,486  | 0,896    |
| <i>AXCS</i> <sub>22</sub>                  | 0,387  | 1,016  | 4,201  | 0,368        | 0,561  | 0,560    |
| <i>AXCS</i> <sub>23</sub>                  | 0,784  | 2,155  | 4,024  | 3,140        | 2,630  | 2,819    |
| <i>AXCS</i> <sub>24</sub>                  | 0,124  | 0,780  | 5,313  | 0,884        | 0,800  | 0,843    |
| <i>AXCS</i> <sub>31</sub>                  | -0,332 | -0,249 | 5,747  | 2,417        | 2,837  | 1,747    |
| <i>AXCS</i> <sub>32</sub>                  | -0,044 | 1,495  | 5,780  | 9,602        | 8,330  | 7,291    |
| <i>AXCS</i> <sub>33</sub>                  | -0,302 | 1,205  | 5,439  | 1,941        | 1,426  | 1,675    |
| <i>AXCS</i> <sub>34</sub>                  | 0,568  | 2,168  | 10,366 | 6,339        | 5,687  | 5,149    |
| Video:RitualDance. TI:28,507 SI:5190,410   |        |        |        |              |        |          |

#### 5.2.2.4 Resolução 2160p (4k)

A última análise por resolução é para vídeos 2160p, ou UHD 4K, cujos resultados estão nas Tabelas 22 e 23. Ao analisar o número de operações de SAD, novamente, percebemos variações significativas na performance desses subtratores nas diferentes etapas do TZS e também nos resultados globais.

A análise dos resultados médios da Soma das Diferenças Absolutas (SAD) por vídeo revela tendências interessantes. No vídeo ParkRunning, observa-se o melhor desempenho geral, com uma média de Total de -0,482. Este resultado demonstra uma tendência de redução nas operações SAD, especialmente nas etapas Dois Pontos (DP) e Busca Refinada (Ref), o que indica uma eficiência com o uso de subtratores aproximados. Já o vídeo Tango tem uma média quase neutra de Total de 0,069, mostrando uma eficiência moderada com pequenas reduções e aumentos nas diversas fases, resultando em um impacto equilibrado. Por outro lado, o vídeo Campfire apresenta uma média de Total de 1,087, onde, apesar de certas fases mostrarem eficiência, o aumento geral nas operações nas outras etapas resulta em uma eficácia reduzida. O vídeo DaylightRoad, com média de Total de 1,743, mostra um aumento consistente nas operações SAD em quase todas as etapas, especialmente na etapa de Busca Refinada (Ref). No vídeo Foodmarket, com uma média Total de 1,924, destaca-se por grandes variações, com eficiências em algumas etapas como Dois Pontos (DP), mas um aumento significativo em outras, como Busca Inicial (BI) e Varredura. Por fim, o vídeo CstRobot, com o pior desempenho, apresenta uma média de Total de 2,581, indicando um aumento substancial nas operações SAD, especialmente na etapa varredura.

Quanto à análise dos resultados médios da Soma das Diferenças Absolutas (SAD) para cada subtrator, com foco no valor agregado total que indica a eficácia global dos subtratores aproximados em comparação ao subtrator *baseline*, destacam-se alguns pontos. O subtrator  $AppS_1$ , equiparado ao subtrator *baseline*, tem um Total de 0 e pode servir como referência. O subtrator  $AXCS1_4$  apresenta o melhor desempenho geral com um Total de -0,091, indicando uma redução efetiva nas operações SAD, sendo especialmente eficiente na Busca Refinada (Ref). Os subtratores  $AppS_2$  e  $AppS_4$  mostram pequenos aumentos no Total, indicando uma eficácia moderada. Os subtratores  $AXCS1_1$ ,  $AXCS2_1$ , e  $AXCS1_2$  apresentam aumentos moderados no Total, com o  $AXCS2_1$  sendo especialmente eficaz na redução de operações SAD na etapa de Busca Refinada (Ref). Os subtratores  $AXCS2_4$ ,  $AXCS1_3$ ,  $AXCS2_2$ ,  $AXCS3_1$ ,  $AXCS2_3$  e  $AXCS3_4$  mostram aumentos ainda maiores no Total, com desempenhos menos eficazes. Os subtratores  $AXCS3_2$  e  $AXCS3_3$  apresentam os piores desempenhos com Totais de 3,102 e 6,821, respectivamente, indicando uma tendência de aumentar significativamente o número de operações SAD em todas as etapas.

A análise por subtrator apresenta que alguns são altamente eficientes em redu-

zir o número de operações SAD, enquanto outros podem degradar significativamente a performance do algoritmo TZS. A etapa Dois Pontos (DP) é onde os subtratores aproximados têm o melhor desempenho, mostrando uma redução significativa nas operações SAD, o que é altamente desejável. Por outro lado, as etapas varredura e Busca Inicial (BI) são particularmente problemáticas, com aumentos substanciais nas operações, o que pode afetar negativamente a eficiência geral do algoritmo TZS.



Tabela 22 – Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações.Resolução 4K

| Subtrator                                             | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
|-------------------------------------------------------|--------|--------|---------|--------------|--------|----------|
| <i>AppS<sub>1</sub></i>                               | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS<sub>2</sub></i>                               | -0,031 | 0,086  | 0,743   | 0,204        | 0,231  | 0,172    |
| <i>AppS<sub>3</sub></i>                               | -0,051 | -0,010 | -0,083  | -0,073       | 0,073  | -0,041   |
| <i>AppS<sub>4</sub></i>                               | 0,096  | 0,033  | -0,513  | 0,279        | -0,238 | 0,158    |
| <i>AXCS<sub>11</sub></i>                              | -0,034 | 0,445  | 1,682   | 0,476        | 0,452  | 0,460    |
| <i>AXCS<sub>12</sub></i>                              | -0,082 | -0,097 | -0,352  | -0,208       | -0,153 | -0,172   |
| <i>AXCS<sub>13</sub></i>                              | 0,379  | 0,995  | 0,914   | 2,510        | 2,235  | 2,049    |
| <i>AXCS<sub>14</sub></i>                              | 0,121  | 0,364  | 1,446   | 0,650        | 0,173  | 0,519    |
| <i>AXCS<sub>21</sub></i>                              | 0,140  | 1,239  | 0,683   | 3,349        | 2,984  | 2,703    |
| <i>AXCS<sub>22</sub></i>                              | 0,145  | 0,372  | -0,942  | 0,535        | 0,624  | 0,496    |
| <i>AXCS<sub>23</sub></i>                              | -0,011 | 0,880  | 0,358   | 1,126        | 1,088  | 1,043    |
| <i>AXCS<sub>24</sub></i>                              | -0,078 | 0,488  | 1,021   | 0,540        | 0,403  | 0,505    |
| <i>AXCS<sub>31</sub></i>                              | -0,058 | 1,782  | -1,626  | 5,033        | 4,388  | 4,024    |
| <i>AXCS<sub>32</sub></i>                              | 0,037  | 0,226  | -1,788  | 0,206        | 0,466  | 0,236    |
| <i>AXCS<sub>33</sub></i>                              | -0,125 | 1,299  | -2,356  | 3,326        | 2,975  | 2,698    |
| <i>AXCS<sub>34</sub></i>                              | 0,060  | 1,184  | 2,237   | 3,144        | 2,803  | 2,543    |
| Video:Campfirebt709videoRange.TI:34,317 e SI:5974,239 |        |        |         |              |        |          |
| Subtrator                                             | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS<sub>1</sub></i>                               | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS<sub>2</sub></i>                               | 1,069  | 1,096  | 1,066   | 1,282        | 0,479  | 1,140    |
| <i>AppS<sub>3</sub></i>                               | 1,477  | 1,350  | 1,236   | 0,964        | 3,098  | 1,380    |
| <i>AppS<sub>4</sub></i>                               | 0,592  | 0,997  | 0,942   | 1,284        | 1,162  | 1,123    |
| <i>AXCS<sub>11</sub></i>                              | 0,452  | 0,925  | 0,701   | 1,274        | -0,470 | 0,943    |
| <i>AXCS<sub>12</sub></i>                              | 0,523  | 1,055  | 0,806   | 2,156        | -0,874 | 1,316    |
| <i>AXCS<sub>13</sub></i>                              | 1,979  | 1,979  | 1,855   | 3,035        | 3,219  | 2,525    |
| <i>AXCS<sub>14</sub></i>                              | 0,701  | 0,536  | 0,872   | 0,949        | 1,205  | 0,768    |
| <i>AXCS<sub>21</sub></i>                              | 0,824  | 1,041  | 1,726   | 2,818        | 0,079  | 1,656    |
| <i>AXCS<sub>22</sub></i>                              | 0,990  | 1,312  | 0,990   | 1,979        | -0,095 | 1,465    |
| <i>AXCS<sub>23</sub></i>                              | 2,269  | 2,187  | 2,437   | 1,811        | 12,924 | 2,967    |
| <i>AXCS<sub>24</sub></i>                              | 1,261  | 0,998  | 1,378   | 2,073        | 2,818  | 1,590    |
| <i>AXCS<sub>31</sub></i>                              | 1,787  | 0,934  | 2,803   | 1,558        | 3,949  | 1,485    |
| <i>AXCS<sub>32</sub></i>                              | 2,809  | 3,343  | 2,583   | 3,647        | 4,008  | 3,561    |
| <i>AXCS<sub>33</sub></i>                              | 2,218  | 1,101  | 4,341   | -1,568       | 38,015 | 3,165    |
| <i>AXCS<sub>34</sub></i>                              | 2,583  | 2,393  | 2,995   | 2,084        | 8,184  | 2,811    |
| Video:DaylightRoad.TI: 10,201 e SI:1592,399           |        |        |         |              |        |          |
| Subtrator                                             | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS<sub>1</sub></i>                               | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS<sub>2</sub></i>                               | -0,004 | 0,074  | -0,970  | -0,206       | -0,607 | -0,165   |
| <i>AppS<sub>3</sub></i>                               | 1,226  | 1,497  | 1,970   | 1,487        | 1,807  | 1,538    |
| <i>AppS<sub>4</sub></i>                               | 0,581  | 0,867  | -0,605  | 0,721        | -0,294 | 0,655    |
| <i>AXCS<sub>11</sub></i>                              | 1,288  | 1,651  | -1,951  | 0,526        | -0,756 | 0,743    |
| <i>AXCS<sub>12</sub></i>                              | 1,943  | 2,266  | -0,311  | 0,562        | 0,462  | 1,104    |
| <i>AXCS<sub>13</sub></i>                              | 1,537  | 2,713  | -3,876  | 1,254        | -1,543 | 1,396    |
| <i>AXCS<sub>14</sub></i>                              | 0,191  | 0,582  | -3,898  | 0,120        | -2,352 | -0,020   |
| <i>AXCS<sub>21</sub></i>                              | 1,642  | 2,498  | -5,474  | 0,191        | -3,366 | 0,520    |
| <i>AXCS<sub>22</sub></i>                              | 2,249  | 3,146  | -0,842  | 2,468        | 1,076  | 2,536    |
| <i>AXCS<sub>23</sub></i>                              | -0,932 | 1,247  | -11,457 | 3,676        | -3,583 | 2,036    |
| <i>AXCS<sub>24</sub></i>                              | 2,408  | 3,633  | -5,452  | 2,100        | -1,600 | 2,169    |
| <i>AXCS<sub>31</sub></i>                              | 2,162  | 4,229  | -9,861  | 2,128        | -5,222 | 1,950    |
| <i>AXCS<sub>32</sub></i>                              | 1,423  | 3,478  | -3,527  | 5,874        | 2,734  | 4,742    |
| <i>AXCS<sub>33</sub></i>                              | 2,870  | 6,268  | -9,669  | 11,604       | 0,948  | 8,691    |
| <i>AXCS<sub>34</sub></i>                              | 0,733  | 2,762  | -13,101 | 4,379        | -3,946 | 2,892    |
| Video:FoodMarket.TI:5,552 e SI:202,701                |        |        |         |              |        |          |
| Subtrator                                             | PM(%)  | BI(%)  | DP(%)   | Varredura(%) | Ref(%) | Total(%) |
| <i>AppS<sub>1</sub></i>                               | 0,000  | 0,000  | 0,000   | 0,000        | 0,000  | 0,000    |
| <i>AppS<sub>2</sub></i>                               | -0,004 | 0,074  | -0,970  | -0,206       | -0,607 | -0,165   |
| <i>AppS<sub>3</sub></i>                               | 1,226  | 1,497  | 1,970   | 1,487        | 1,807  | 1,538    |
| <i>AppS<sub>4</sub></i>                               | 0,581  | 0,867  | -0,605  | 0,721        | -0,294 | 0,655    |
| <i>AXCS<sub>11</sub></i>                              | 1,288  | 1,651  | -1,951  | 0,526        | -0,756 | 0,743    |
| <i>AXCS<sub>12</sub></i>                              | 1,943  | 2,266  | -0,311  | 0,562        | 0,462  | 1,104    |
| <i>AXCS<sub>13</sub></i>                              | 1,537  | 2,713  | -3,876  | 1,254        | -1,543 | 1,396    |
| <i>AXCS<sub>14</sub></i>                              | 0,191  | 0,582  | -3,898  | 0,120        | -2,352 | -0,020   |
| <i>AXCS<sub>21</sub></i>                              | 1,642  | 2,498  | -5,474  | 0,191        | -3,366 | 0,520    |
| <i>AXCS<sub>22</sub></i>                              | 2,249  | 3,146  | -0,842  | 2,468        | 1,076  | 2,536    |
| <i>AXCS<sub>23</sub></i>                              | -0,932 | 1,247  | -11,457 | 3,676        | -3,583 | 2,036    |
| <i>AXCS<sub>24</sub></i>                              | 2,408  | 3,633  | -5,452  | 2,100        | -1,600 | 2,169    |
| <i>AXCS<sub>31</sub></i>                              | 2,162  | 4,229  | -9,861  | 2,128        | -5,222 | 1,950    |
| <i>AXCS<sub>32</sub></i>                              | 1,423  | 3,478  | -3,527  | 5,874        | 2,734  | 4,742    |
| <i>AXCS<sub>33</sub></i>                              | 2,870  | 6,268  | -9,669  | 11,604       | 0,948  | 8,691    |
| <i>AXCS<sub>34</sub></i>                              | 0,733  | 2,762  | -13,101 | 4,379        | -3,946 | 2,892    |





### 5.2.2.5 Análise do Resultado do Cálculo do SAD para blocos 32x32

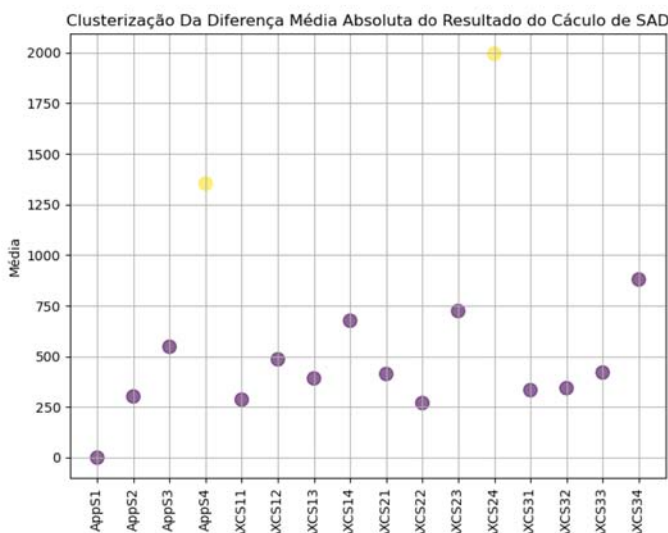
A Tabela 24 apresenta os resultados da diferença média absoluta dos valores de SAD entre os subtratores aproximados e o subtrator *baseline*, considerando todas as sequências de todas as resoluções. Essa análise revela aspectos significativos do desempenho dos subtratores analisados. Os valores variaram entre 1.993,76, alcançado pelo  $AXCS_{24}$ , e 0,00, para o  $AppS_1$ . A média das diferenças é de 588,58 com um desvio padrão de 484,90, sublinhando a variabilidade entre os subtratores. Como esperado, existe a tendência de ampliação da diferença a partir da ampliação do grau de aproximação.

Tabela 24 – Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32

| Subtrator   | Média     | Subtrator   | Média   |
|-------------|-----------|-------------|---------|
| $AppS_1$    | 0,000     | $AXCS_{11}$ | 286,530 |
| $AppS_2$    | 301,960   | $AXCS_{12}$ | 485,330 |
| $AppS_3$    | 547,370   | $AXCS_{13}$ | 391,190 |
| $AppS_4$    | 1.352,980 | $AXCS_{14}$ | 675,960 |
| Subtrator   | Média     | Subtrator   | Média   |
| $AXCS_{21}$ | 412,870   | $AXCS_{31}$ | 333,240 |
| $AXCS_{22}$ | 269,630   | $AXCS_{32}$ | 342,970 |
| $AXCS_{23}$ | 723,910   | $AXCS_{33}$ | 420,170 |
| $AXCS_{24}$ | 1.993,760 | $AXCS_{34}$ | 879,410 |

A análise do gráfico na Figura 37 revela a formação de dois clusters distintos em termos de diferença média absoluta. O *cluster* visualizado em roxo, inclui a maioria dos subtratores e exibe diferenças menores, o que sugere aproximações mais precisas e confiáveis. Em contrapartida, o *cluster*, em amarelo, composto por apenas dois subtratores, mostra as maiores diferenças, indicando cálculos menos precisos e uma maior divergência em relação ao valor exato.

Figura 37 – Classificação dos Subtratores por Resultado do Cálculo de SAD. Blocos 32x32



A partir da Figura 37 é possível perceber que o subtrator *AppS* apresentou resultados intermediários nesta avaliação, apresentando resultados melhores do que o subtrator *AXCS2* e piores do que os subtratores *AXCS1* e *AXCS3*. Por outro lado, considerando os níveis com um e dois bits aproximados, o subtrator *AppS* atingiu resultados melhores do que os demais operadores, exceto para o *AXCS2<sub>2</sub>*, que teve um resultado um pouco melhor do que o *AppS<sub>2</sub>*.

#### 5.2.2.6 Análise da Distância dos Vetores de Movimento para blocos 32x32

Na Tabela 25 estão apresentados os resultados da distância média absoluta dos valores dos vetores de movimento gerados pelo TZS usando os subtratores aproximados e o *baseline*. A média desses valores é 0,424, com desvio padrão de 0,503. Estes valores indicam, por um lado, que o uso dos subtratores aproximados tende a gerar vetores de movimento que apontam para regiões próximas da região ótima do vídeo. Por outro, o desvio padrão destaca uma variabilidade significativa entre os resultados, revelando que, apesar de a média sugerir uma consistência, as medidas individuais dos subtratores variam consideravelmente. Observando os resultados da Tabela 25 é possível perceber um resultado esperado, qual seja, que a ampliação no número de bits aproximado amplia a diferença nos resultados dos vetores gerados. De todo modo, a diferença média é superior a 1% em apenas dois casos.

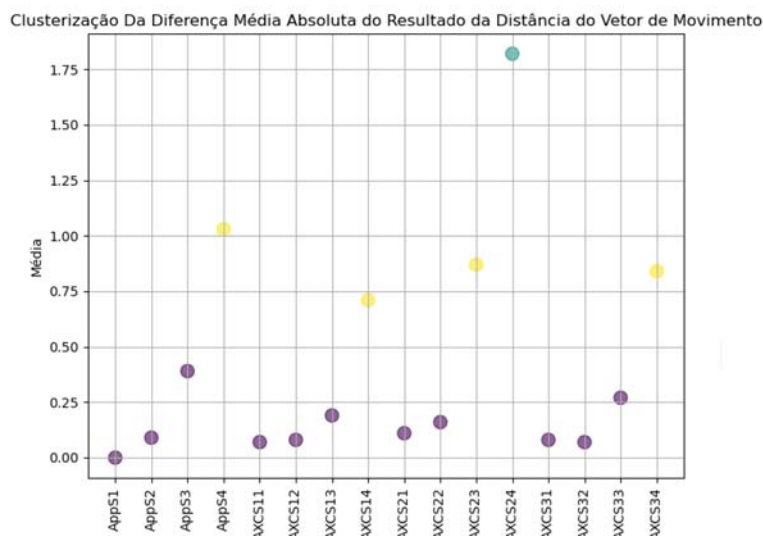
Tabela 25 – Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32

| Subtrator                | Média | Subtrator                | Média |
|--------------------------|-------|--------------------------|-------|
| <i>AppS<sub>1</sub></i>  | 0,000 | <i>AXCS1<sub>1</sub></i> | 0,070 |
| <i>AppS<sub>2</sub></i>  | 0,090 | <i>AXCS1<sub>2</sub></i> | 0,080 |
| <i>AppS<sub>3</sub></i>  | 0,390 | <i>AXCS1<sub>3</sub></i> | 0,190 |
| <i>AppS<sub>4</sub></i>  | 1,030 | <i>AXCS1<sub>4</sub></i> | 0,710 |
| Subtrator                | Média | Subtrator                | Média |
| <i>AXCS2<sub>1</sub></i> | 0,110 | <i>AXCS3<sub>1</sub></i> | 0,080 |
| <i>AXCS2<sub>2</sub></i> | 0,160 | <i>AXCS3<sub>2</sub></i> | 0,070 |
| <i>AXCS2<sub>3</sub></i> | 0,870 | <i>AXCS3<sub>3</sub></i> | 0,270 |
| <i>AXCS2<sub>4</sub></i> | 1,820 | <i>AXCS3<sub>4</sub></i> | 0,840 |

A Figura 38 apresenta a clusterização dos resultados anteriores. Nesta figura é possível perceber que o maior *cluster*, é composto por subtratores que apresentam as menores diferenças médias absolutas, sugerindo cálculos mais precisos. O segundo *cluster* inclui subtratores com diferenças médias absolutas moderadas, indicando uma precisão inferior à do primeiro *cluster*, mas ainda dentro de um limite aceitável. Já o terceiro *cluster* contém apenas um subtrator, que se destaca por uma diferença média absoluta significativamente maior que os demais.

Nesta análise, novamente o uso do subtrator *AppS* atingiu resultados intermediários, com resultados melhores do que um dos subtratores avaliados e piores do que os resultados dos outros dois. Ainda assim os valores menores, sendo levemente

Figura 38 – Classificação dos Vetores de Movimento por Subtrator Aproximado. Blocos 32x32



superior a 1% em apenas um caso.

#### 5.2.2.7 Eficiência de Codificação para blocos 32x32

A análise Tabela 26 apresenta os impactos em eficiência de codificação para uso dos subtratores aproximados em blocos de tamanho 32x32. Estes resultados revelam variações notáveis na eficiência de codificação com o uso de diferentes subtratores, com vários resultados negativos, ou seja, apontando para uma elevação na eficiência de codificação. Mas a observação mais importante é que, nos casos em que há perdas na eficiência de codificação, todos os valores encontrados foram extremamente baixos, sempre inferiores a 0,5% no BD-Rate. Apenas quatro resultados ficaram acima dos 0,3%, com o máximo impacto encontrado sendo 0,434%.

O subtrator *AppS* apresentou resultados muito bons neste critério, com máximo impacto encontrado para o *AppS<sub>4</sub>* no vídeo SlideShow, com um BD-Rate de 0,178%. Além disso, considerando as 68 avaliações apresentadas para o *AppS*, 17 apresentaram impacto nulo em BD-Rate, e 26 apresentaram valores negativos, ou seja, com ganhos na eficiência de codificação.

Concluindo, a avaliação dos subtratores aproximados apontou que é impossível indicar qual é o subtrator mais indicado para uso no cálculo do SAD no TZS, uma vez que os resultados atingidos apontam para uma grande variação em função das características do vídeo sendo processado. Por outro lado, é importante ressaltar que o uso de diferentes subtratores aproximados, mesmo com uma aproximação no cálculo do SAD e com a geração de vetores de movimento levemente distintos, o impacto em eficiência de codificação é mínimo, indicando que o uso de subtratores aproximados no cálculo do SAD no TZS é uma opção promissora para reduzir área, potência e energia em implementações em hardware do TZS.

Tabela 26 – Eficiência BD-RATE. Blocos 32x32

| Resolução | Vídeo           | $AppS_1$ | $AppS_2$ | $AppS_3$ | $AppS_4$ | $AXCS1_1$ | $AXCS1_2$ | $AXCS1_3$ | $AXCS1_4$ |
|-----------|-----------------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|
| 480       | BasketballDrill | 0,000    | 0,129    | 0,032    | -0,036   | 0,024     | 0,088     | -0,090    | 0,020     |
| 480       | BQMall          | 0,000    | -0,045   | 0,001    | -0,357   | -0,060    | -0,006    | -0,148    | -0,008    |
| 480       | PartyScene      | 0,000    | 0,016    | 0,089    | 0,150    | 0,035     | 0,090     | 0,008     | -0,011    |
| 480       | RaceHorsesC     | 0,000    | -0,056   | -0,037   | 0,038    | 0,019     | -0,061    | -0,008    | -0,069    |
| 720       | FourPeople      | 0,000    | -0,088   | -0,024   | 0,075    | 0,017     | -0,118    | -0,002    | 0,046     |
| 720       | Johnny          | 0,000    | -0,126   | 0,131    | 0,114    | 0,042     | -0,099    | -0,064    | 0,118     |
| 720       | KristenAndSara  | 0,000    | 0,072    | 0,075    | 0,039    | 0,065     | 0,102     | 0,157     | 0,142     |
| 720       | SlideEditing    | 0,000    | -0,113   | -0,047   | 0,160    | 0,053     | -0,140    | 0,006     | -0,090    |
| 720       | SlideShow       | 0,000    | -0,598   | 0,015    | 0,178    | -0,526    | 0,128     | 0,341     | -0,626    |
| 1080      | BasketballDrive | 0,000    | -0,061   | 0,119    | -0,054   | 0,141     | 0,269     | 0,278     | 0,238     |
| 1080      | Cactus          | 0,000    | 0,036    | 0,060    | -0,067   | 0,052     | 0,048     | 0,085     | 0,258     |
| 1080      | MarketPlace     | 0,000    | 0,038    | -0,020   | -0,159   | 0,012     | 0,219     | -0,081    | 0,035     |
| 1080      | RitualDance     | 0,000    | -0,018   | 0,134    | 0,013    | 0,223     | 0,267     | 0,021     | 0,139     |
| 2160      | CatRobotjvet    | 0,000    | 0,026    | 0,049    | -0,012   | 0,030     | 0,068     | 0,021     | 0,042     |
| 2160      | FoodMarket4     | 0,000    | -0,116   | -0,119   | -0,053   | -0,237    | -0,009    | -0,091    | 0,071     |
| 2160      | ParkRunning3    | 0,000    | -0,004   | -0,013   | 0,010    | 0,000     | -0,007    | -0,034    | 0,205     |
| 2160      | Tango2          | 0,000    | -0,153   | -0,087   | -0,125   | -0,172    | -0,268    | -0,100    | -0,144    |

| Resolução | Vídeo           | $AXCS2_1$ | $AXCS2_2$ | $AXCS2_3$ | $AXCS2_4$ | $AXCS3_1$ | $AXCS3_2$ | $AXCS3_3$ | $AXCS3_4$ |
|-----------|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 480       | BasketballDrill | 0,210     | 0,053     | 0,077     | 0,159     | 0,227     | -0,057    | -0,052    | 0,222     |
| 480       | BQMall          | -0,160    | -0,070    | -0,010    | -0,138    | -0,076    | -0,066    | -0,125    | 0,001     |
| 480       | PartyScene      | 0,000     | 0,011     | 0,070     | 0,011     | 0,058     | 0,154     | 0,011     | 0,026     |
| 480       | RaceHorsesC     | 0,162     | 0,049     | 0,067     | 0,166     | 0,166     | 0,079     | 0,075     | -0,058    |
| 720       | FourPeople      | -0,077    | 0,079     | -0,032    | 0,034     | -0,060    | -0,101    | 0,071     | -0,463    |
| 720       | Johnny          | 0,106     | 0,133     | -0,009    | 0,434     | 0,163     | -0,037    | -0,195    | -0,040    |
| 720       | KristenAndSara  | 0,123     | -0,070    | -0,046    | 0,179     | 0,066     | 0,259     | 0,048     | -0,014    |
| 720       | SlideEditing    | -0,148    | 0,000     | -0,139    | -0,109    | 0,039     | -0,086    | 0,029     | -0,202    |
| 720       | SlideShow       | 0,113     | -0,110    | 0,191     | 0,358     | 0,324     | -0,057    | -0,272    | -0,008    |
| 1080      | BasketballDrive | 0,103     | -0,055    | -0,309    | -0,005    | 0,049     | 0,057     | -0,130    | 0,227     |
| 1080      | Cactus          | 0,108     | 0,080     | 0,036     | 0,108     | 0,011     | 0,016     | 0,085     | -0,063    |
| 1080      | MarketPlace     | -0,054    | 0,005     | 0,018     | 0,012     | -0,096    | -0,041    | 0,055     | 0,001     |
| 1080      | RitualDance     | 0,159     | 0,008     | 0,174     | 0,048     | 0,036     | 0,158     | -0,020    | 0,021     |
| 2160      | CatRobotjvet    | 0,018     | -0,023    | 0,039     | 0,035     | 0,050     | 0,006     | -0,076    | -0,062    |
| 2160      | FoodMarket4     | -0,055    | -0,036    | 0,007     | 0,050     | -0,192    | -0,092    | -0,204    | -0,053    |
| 2160      | ParkRunning3    | -0,029    | -0,007    | -0,047    | -0,011    | -0,003    | -0,001    | 0,010     | -0,047    |
| 2160      | Tango2          | -0,128    | -0,154    | -0,079    | -0,081    | -0,140    | -0,092    | -0,022    | -0,210    |

### 5.3 Sumarização de todos os Resultados:Desempenho Geral dos Subtratores

Nesta sessão, compilamos todos os resultados apresentados até o momento, incluindo aqueles encontrados no apêndice - B Análise para Blocos de Tamanhos Fixo. O total de operações de SAD resume o volume bruto dessas operações, seguindo a mesma metodologia aplicada anteriormente: a relação entre o subtrator aproximado e o *baseline*. Além disso, os resultados de SAD, vetor de movimento e eficiência de codificação foram sintetizados a partir das médias dos dados apresentados nas tabelas anteriores.

#### 5.3.1 Totais de SADs Executados

Esta análise, apresentada na Tabela 27, traz a média de todos resultados, abrangendo dados de blocos de tamanhos 16x16, 32x32, 64x64 e 128x128. É possível perceber que o uso de subtratores aproximados causou diferentes impactos nas diferentes etapas do TZS. Alguns operadores geraram uma redução no número de operações no cálculo do Preditor Mediano (PM), outros conseguiram reduzir os cálculos

de SAD na etapa de Dois Pontos e outros ainda conseguiram reduzir as operações na etapa de Busca Refinada (Ref). A destacar que nenhum operador conseguiu apresentar resultados médios de redução no número de operações na etapa de Busca Inicial (BI) e na etapa de Varredura. Os resultados totais apontam que quase todos os subtratores nas configurações avaliadas trazem resultados de ampliação no número de cálculos de SAD, variando de 0% a 2,856%

Em termos de resultados por etapa, o subtrator  $AXCS1_1$  alcançou a maior redução 0,274% no número de operações de SAD na etapa de Preditor Mediano (PM). Em contraste, o  $AXCS1_3$  apresentou o pior resultado, com um aumento de 0,826%, sugerindo menor eficiência. Na etapa de Busca Inicial (BI), o subtrator  $AppS_1$  registrou um desempenho equivalente ao *baseline*, enquanto o  $AXCS3_3$  mostrou um aumento de 1,392%, sendo o pior caso. Na etapa de Dois Pontos (DP) o melhor caso ficou com o  $AXCS3_4$ , com -1,322%, enquanto o pior caso ficou com o  $AXCS1_4$ , com 0,098%. Na etapa de Varredura, novamente o  $AppS_1$  atingiu o melhor resultado, enquanto o pior resultado ficou com o  $AXSC3_3$ , com uma ampliação expressiva de 4,128% no número de operações. Por fim, na etapa de Busca Refinada (Ref), o melhor caso ficou com o  $AppS_4$  com uma redução de 0,168%, e pior caso ficou novamente com o  $AXSC3_3$ , com uma ampliação de 3,070%

Tabela 27 – Sumário da Contagem de SAD para Subtratores Aproximados em Diferentes Tamanhos de Blocos

| Subtrator | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|-----------|---------------------|------------------|----------------|--------------|-------------|----------|
| $AppS_1$  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| $AppS_2$  | -0,014              | 0,096            | 0,021          | 0,075        | -0,059      | 0,070    |
| $AppS_3$  | -0,183              | 0,108            | 0,040          | 0,083        | 0,204       | 0,099    |
| $AppS_4$  | 0,510               | 0,585            | -0,219         | 0,412        | -0,168      | 0,430    |
| $AXCS1_1$ | -0,274              | 0,292            | 0,010          | 0,226        | -0,146      | 0,210    |
| $AXCS1_2$ | -0,209              | 0,009            | -0,172         | 0,069        | -0,150      | 0,019    |
| $AXCS1_3$ | 0,399               | 0,592            | -0,087         | 1,198        | 0,435       | 0,867    |
| $AXCS1_4$ | 0,255               | 0,427            | 0,098          | 0,556        | -0,157      | 0,433    |
| $AXCS2_1$ | 0,043               | 0,444            | -0,823         | 1,228        | 0,336       | 0,805    |
| $AXCS2_2$ | 0,448               | 0,533            | -0,084         | 0,581        | 0,109       | 0,515    |
| $AXCS2_3$ | 0,511               | 0,986            | -0,719         | 1,432        | 0,607       | 1,156    |
| $AXCS2_4$ | 0,390               | 1,009            | -0,314         | 1,734        | 0,721       | 1,321    |
| $AXCS3_1$ | 0,443               | 0,849            | -1,494         | 2,357        | 0,879       | 1,571    |
| $AXCS3_2$ | -0,128              | 0,722            | -0,390         | 2,174        | 1,248       | 1,457    |
| $AXCS3_3$ | 0,826               | 1,392            | -1,222         | 4,128        | 3,070       | 2,856    |
| $AXCS3_4$ | 0,691               | 1,257            | -1,322         | 2,514        | 1,191       | 1,845    |

O  $AppS_1$  teve o melhor resultado Total entre todos os subtratores, não alterando a quantidade de SADs executados.

### 5.3.2 Média Geral dos Resultados de SAD

A Tabela 28 apresenta a média geral da diferença absoluta dos resultados do SAD discutidos anteriormente. A média geral das diferenças de todos subtratores é de 2.089,34. Esse valor atua como um indicador central, representando o nível médio de atividade ou o resultado médio entre todos os subtratores analisados. Já o desvio padrão, que é de 1.561,80, mede a variabilidade dos resultados entre os subtrato-



res. Conforme esperando, com a ampliação do número de bits aproximados há uma aplicação no nível da imprecisão.

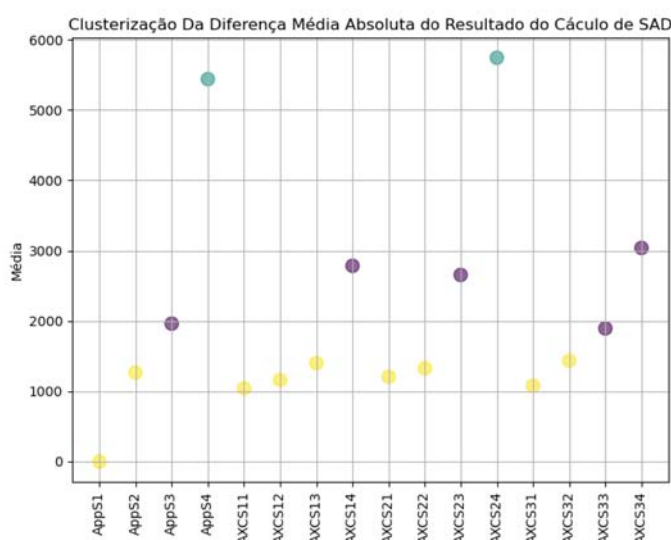
Tabela 28 – Sumarização da Diferença Absoluta do Resultado de SAD entre o Subtrator Aproximado e o Subtrator Baseline

| Subtrator | Média    | Subtrator | Média    |
|-----------|----------|-----------|----------|
| $AppS_1$  | 0,00     | $AXCS1_1$ | 1.043,43 |
| $AppS_2$  | 1.265,17 | $AXCS1_2$ | 1.160,08 |
| $AppS_3$  | 1.961,74 | $AXCS1_3$ | 1.400,65 |
| $AppS_4$  | 5.437,78 | $AXCS1_4$ | 2.783,95 |
| Subtrator | Média    | Subtrator | Média    |
| $AXCS2_1$ | 1.205,98 | $AXCS3_1$ | 1.080,42 |
| $AXCS2_2$ | 1.326,79 | $AXCS3_2$ | 1.436,09 |
| $AXCS2_3$ | 2.654,38 | $AXCS3_3$ | 1.893,91 |
| $AXCS2_4$ | 5.741,43 | $AXCS3_4$ | 3.037,70 |

O subtrator  $AppS_1$ , com uma média igual a zero, representa o melhor resultado. Por outro lado, o  $AXCS2_4$  destaca-se como o pior resultado, apresentando o maior valor de média de diferenças.

Os dados apresentados na Figura 39 estão organizados em três *clusters* distintos, cada um representado por uma cor específica no gráfico. Esses *clusters* agrupam subtratores aproximados que exibem desempenhos similares quando comparados com o subtrator *baseline*, com base na média da diferença absoluta dos resultados.

Figura 39 – Classificação dos Subtratores pela Média das Diferenças Absolutas dos Resultado do Cálculo de SAD.



O primeiro *cluster* (apresentado em verde), inclui subtratores cuja média de diferença absoluta é a mais elevada. O segundo *cluster* (identificado pela cor roxa) abrange subtratores com um desempenho intermediário, ocupando uma posição entre os extremos de precisão. Por fim, o terceiro *cluster* compreende subtratores cujas diferenças médias absolutas são as menores, ou seja, indica os melhores resultados.

Analisando apenas o *AppS*, é possível perceber que a tendência apontada nas análises similares das seções anteriores se repete aqui, ou seja, este operador ocupa uma posição intermediária entre os subtratores avaliados. O *AppS* apresenta o melhor resultado para a aproximação de um bit, o segundo melhor resultado para a aproximação de dois bits e o terceiro melhor resultado para aproximação de três e quatro bits.

### 5.3.3 Distância dos Vetores de Movimento

A Tabela 29 condensa os resultados da distância dos vetores de movimento gerados com os subtratores aproximados e os vetores de movimentos gerados com o subtrator *baseline*. A média, geral foi de 0,54 e o desvio padrão foi de 0,62. Naturalmente, e como esperado, quanto maior o número de bits aproximados, maior é a distância dos vetores de movimento. Mas destaca-se que mesmo nos níveis de imprecisão mais elevados, o valor médio da distância não foi superior a 2,2425, indicando um elevado nível de precisão. Nos melhores casos, a distância ficou em zero ou próxima de zero. O subtrator *AppS*<sub>1</sub>, com média igual a 0, destaca-se como o melhor operador. Por outro lado, o *AXCS*<sub>24</sub>, apresentando a maior média entre todos, é identificado como o de pior desempenho neste critério.

Tabela 29 – Sumarização da Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para Blocos 32x32

| Subtrator                 | Média  | Subtrator                 | Média  |
|---------------------------|--------|---------------------------|--------|
| <i>AppS</i> <sub>1</sub>  | 0,000  | <i>AXCS</i> <sub>11</sub> | 0,0725 |
| <i>AppS</i> <sub>2</sub>  | 0,135  | <i>AXCS</i> <sub>12</sub> | 0,0875 |
| <i>AppS</i> <sub>3</sub>  | 0,53   | <i>AXCS</i> <sub>13</sub> | 0,27   |
| <i>AppS</i> <sub>4</sub>  | 1,2125 | <i>AXCS</i> <sub>14</sub> | 0,9125 |
| Subtrator                 | Média  | Subtrator                 | Média  |
| <i>AXCS</i> <sub>21</sub> | 0,09   | <i>AXCS</i> <sub>31</sub> | 0,0925 |
| <i>AXCS</i> <sub>22</sub> | 0,205  | <i>AXCS</i> <sub>32</sub> | 0,125  |
| <i>AXCS</i> <sub>23</sub> | 1,2025 | <i>AXCS</i> <sub>33</sub> | 0,405  |
| <i>AXCS</i> <sub>24</sub> | 2,2425 | <i>AXCS</i> <sub>34</sub> | 1,0575 |

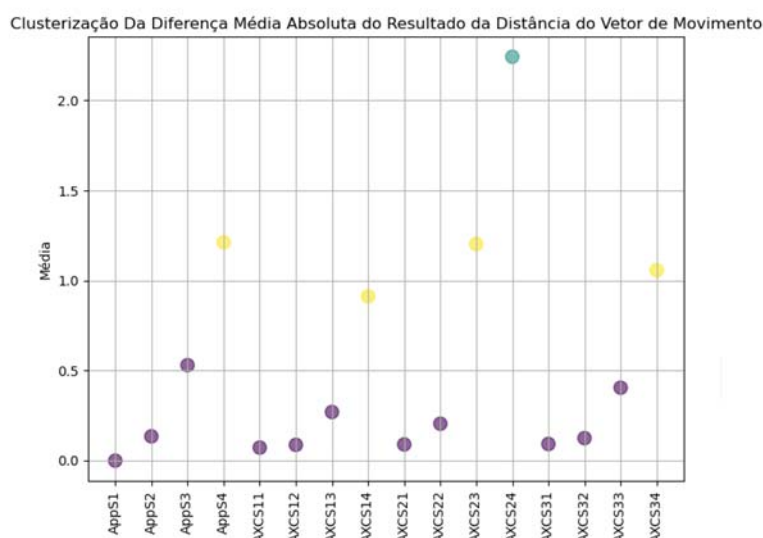
Estes resultados reafirmam a robustez dos subtratores aproximados, uma vez que a consistência observada nos valores de SAD está diretamente ligada à precisão na determinação dos vetores de movimento. Em outras palavras, mesmo diante de variações nos resultados de SAD, a orientação dos vetores de movimento mantém-se inalterada. Isso sugere que os subtratores aproximados são uma ferramenta valiosa para a análise de movimento, garantindo que a direção correta seja identificada independentemente das especificidades do bloco utilizado.

Ao analisar o gráfico apresentado na Figura 40, é possível identificar que os subtratores podem ser categorizados novamente em três grupos distintos. No primeiro grupo, identificado pela cor roxa, o uso dos subtratores aproximado gera vetores de movimento muito próximos aos encontrados pelo subtrator *baseline*. Em seguida, o



grupo intermediário, marcado pela cor amarela, inclui subtratores cujos vetores de movimento apresentam uma discrepância um pouco maior em relação ao vetor exato. Por fim, o grupo na cor verde destaca-se por representar o subtrator com a maior diferença entre os vetores de movimento, indicando uma discrepância mais significativa em relação ao subtrator exato.

Figura 40 – Classificação da Média de todos Vetores de Movimento por Subtrator Aproximado



Na análise do subtrator *AppS*, neste quesito o comportamento é similar aos anteriores, ou seja, ele ocupa uma posição intermediária entre os subtratores avaliados. O *AppS* apresenta o melhor resultado para a aproximação de um bit, o terceiro melhor resultado para a aproximação de dois, três e quatro bits.

### 5.3.4 Eficiência de Codificação

A Tabela 30 sumariza os resultados da eficiência de codificação. Os resultados de BD-Rate mostram uma variação inexpressiva com o uso dos subtratores aproximados, sempre com pelo menos um zero depois da vírgula. Alguns resultados apresentam ganhos (resultados negativos) e outros apresentam perdas (resultados positivos), mas todos muito próximos a zero. No total, sete dos casos avaliados geraram ganhos e nove geraram perdas. O máximo ganho foi de 0,047, nos subtratores *AppS*<sub>2</sub> e *AXCS*<sub>34</sub>. Por outro lado, a perda máxima foi de 0,060, no subtrator *AXCS*<sub>24</sub>. Esses resultados indicam que o uso de subtratores aproximados, de modo geral, não trazem impactos relevantes na eficiência de codificação, demonstrando um desempenho muito bom neste critério que, de fato, é o mais relevante dentre os apresentados até aqui nesta seção do texto.

Neste contexto, o subtrator *AppS* atingiu resultados muito bons, trazendo ganhos em eficiência de codificação em duas das quatro configurações avaliadas. Ele tem o melhor resultado para dois bits aproximados, o segundo melhor resultado para um bit e

Tabela 30 – Eficiência Média Total de Codificação em BD-Rate

| Subtrator                | Média  | - | Subtrator                 | Média  |
|--------------------------|--------|---|---------------------------|--------|
| <i>AppS</i> <sub>1</sub> | 0,000  | - | <i>AXCS</i> <sub>11</sub> | -0,015 |
| <i>AppS</i> <sub>2</sub> | -0,047 | - | <i>AXCS</i> <sub>12</sub> | 0,025  |
| <i>AppS</i> <sub>3</sub> | 0,011  | - | <i>AXCS</i> <sub>13</sub> | 0,012  |
| <i>AppS</i> <sub>4</sub> | -0,004 | - | <i>AXCS</i> <sub>14</sub> | 0,006  |

| Subtrator                 | Média  | - | Subtrator                 | Média  |
|---------------------------|--------|---|---------------------------|--------|
| <i>AXCS</i> <sub>21</sub> | 0,029  | - | <i>AXCS</i> <sub>31</sub> | 0,029  |
| <i>AXCS</i> <sub>22</sub> | -0,008 | - | <i>AXCS</i> <sub>22</sub> | 0,009  |
| <i>AXCS</i> <sub>23</sub> | -0,005 | - | <i>AXCS</i> <sub>33</sub> | -0,046 |
| <i>AXCS</i> <sub>24</sub> | 0,060  | - | <i>AXCS</i> <sub>34</sub> | -0,047 |

para quatro bits aproximados e o terceiro melhor resultado para três bits aproximados.

Tabela 31 – Comparação entre o Total de SADS Executados e a Eficiência de Codificação em BD-Rate

| Subtrator                 | QtdSAD             | QtdSAD% | BDRate |
|---------------------------|--------------------|---------|--------|
| <i>AppS</i> <sub>1</sub>  | 25.726.762.029,240 | 0,000   | 0,000  |
| <i>AppS</i> <sub>2</sub>  | 25.744.761.619,991 | 0,070   | -0,054 |
| <i>AppS</i> <sub>3</sub>  | 25.752.309.052,523 | 0,099   | 0,015  |
| <i>AppS</i> <sub>4</sub>  | 25.837.338.984,126 | 0,430   | -0,007 |
| <i>AXCS</i> <sub>11</sub> | 25.780.715.695,948 | 0,210   | -0,026 |
| <i>AXCS</i> <sub>12</sub> | 25.731.738.721,104 | 0,019   | 0,019  |
| <i>AXCS</i> <sub>13</sub> | 25.949.774.236,581 | 0,867   | 0,006  |
| <i>AXCS</i> <sub>14</sub> | 25.838.139.287,260 | 0,433   | 0,010  |
| <i>AXCS</i> <sub>21</sub> | 25.933.781.865,136 | 0,805   | 0,018  |
| <i>AXCS</i> <sub>22</sub> | 25.859.231.366,025 | 0,515   | -0,011 |
| <i>AXCS</i> <sub>23</sub> | 26.024.164.268,136 | 1,156   | -0,013 |
| <i>AXCS</i> <sub>24</sub> | 26.066.673.974,993 | 1,321   | 0,057  |
| <i>AXCS</i> <sub>31</sub> | 26.130.990.617,398 | 1,571   | 0,027  |
| <i>AXCS</i> <sub>32</sub> | 26.101.657.113,462 | 1,457   | -0,001 |
| <i>AXCS</i> <sub>33</sub> | 26.461.497.871,352 | 2,856   | -0,045 |
| <i>AXCS</i> <sub>34</sub> | 26.201.518.636,639 | 1,845   | -0,048 |

Uma avaliação adicional é apresentada na Tabela 31, incluindo a quantidade total de SADs executados com o uso de cada subtrator, o percentual do aumento no número de SADs em relação ao subtrator *baseline* e a eficiência de codificação. As variações no número dessas operações demonstram que, mesmo com diferenças significativas na quantidade de cálculos, os subtratores aproximados tendem a apresentar apenas pequenas variações na métrica BD-Rate em comparação com o *baseline*. Isso sugere que alterações na quantidade de cálculos não comprometem necessariamente a eficiência da codificação de vídeo, o que é um indicativo de que a robustez do sistema mantém-se apesar das aproximações implementadas.

Esses resultados reforçam a ideia de que o uso de subtratores aproximados é uma abordagem eficaz para otimizar o desempenho da codificação de vídeo. A manutenção da eficiência de codificação, apesar das variações no número de SADs, permite que ajustes sejam realizados de acordo com as necessidades específicas de cada aplicação. Isso sugere que as alterações na quantidade de operações SAD não

implicam diretamente em alterações significativas na qualidade, proporcionando um caminho viável para a otimização sem grandes sacrifícios.

No próximo capítulo, iremos aprofundar nossa análise, focando nos subtratores aproximados embarcados em uma arquitetura elaborada de forma conceitual especificamente para testar esses subtratores. Essa análise detalhada permitirá uma compreensão mais completa sobre como as variações nas operações SAD influenciam o desempenho da codificação em cenários controlados e altamente especializados.

## 6 ARQUITETURA DO TZS

Neste capítulo, apresentamos a arquitetura do TZS, cuja visão global está ilustrada na Figura 41. Essa estrutura foi concebida para suportar uma ampla variedade de configurações do TZS, conforme discutido nas seções anteriores. A arquitetura se divide em quatro módulos principais: Controle, Posições, Corte e SAD. Esta modularidade não apenas aumenta a versatilidade do sistema, mas também facilita a compreensão e a expansão futura do mesmo.

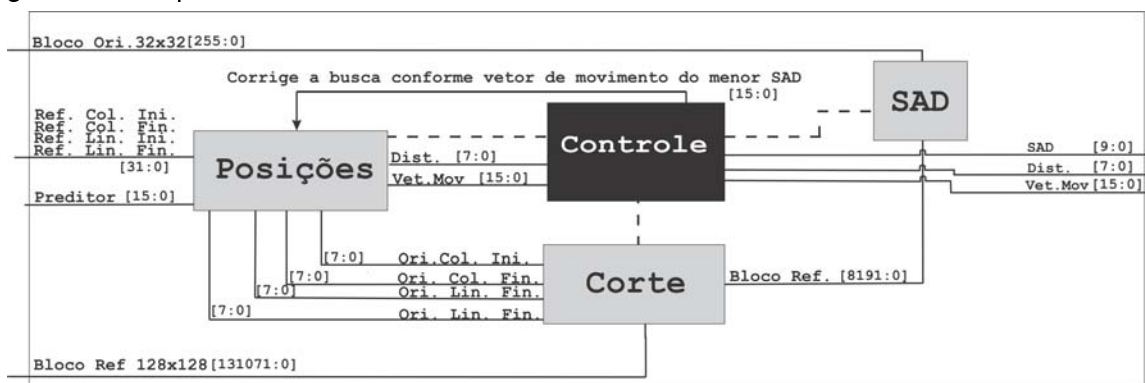
Uma característica destacada dessa arquitetura é sua capacidade de processar blocos de  $32 \times 32$  em uma área de busca de  $128 \times 128$ . Essa funcionalidade permite um tratamento eficiente das informações visuais, essencial para a aplicação prática do TZS.

A arquitetura que desenvolvemos para o TZS é composta por quatro entradas específicas, cada uma desempenhando um papel crucial no processo. A primeira entrada é a área de busca, que contém  $128 \times 128$  amostras, cada uma com 8 bits. Embora não seja viável uma interface de memória com essa largura, para fins de nosso estudo, abstraímos esse aspecto, assumindo a disponibilidade dos dados. Este estudo não foca em técnicas de reuso de dados, portanto, essa discussão está fora de nosso escopo.

A segunda entrada consiste em uma linha do bloco atual, que inclui 32 amostras, também com 8 bits cada. Essas amostras são essenciais para o cálculo do SAD (Sum of Absolute Differences), e para isso, presumimos a existência de um buffer externo que gerencia essas amostras. Segue-se a terceira entrada, que são as coordenadas do bloco de referência. Essas coordenadas são vitais para o módulo de posições, responsável pelo cálculo do SAD. Este processo é crucial para a precisão do sistema. Por fim, a quarta entrada é o preditor Mediano de 16 bits, uma concatenação de segmentos de 8 bits. Esta entrada é fundamental para a integração e sincronização dos dados processados. Essa configuração robusta permite que nosso sistema processe eficientemente as informações, garantindo alta performance e precisão nos resultados.

A seguir, detalharemos os quatro módulos fundamentais que compõem a arquitetura.

Figura 41 – Arquitetura do TZS



tura em foco. Ao final, explicaremos o funcionamento geral dessa arquitetura. Essa explanação será acompanhada por uma análise abrangente, que avaliará os resultados desses módulos em termos de frequência, potência, energia e área. Cada módulo será abordado individualmente para elucidar tanto sua importância quanto seu funcionamento no contexto global da arquitetura.

## 6.1 Metodologia de Síntese

O processo de síntese foi realizado com o auxílio de duas ferramentas avançadas da Cadence: o *Incisive*, versão 9.20, utilizado para a simulação do comportamento dos circuitos, e o *RTL Compiler*, versão 11.10, empregado para a compilação e otimização dos códigos HDL. A tecnologia utilizada para os gates foi a de 40 nanômetros a 0,9 Volts, fornecida pela TSMC.

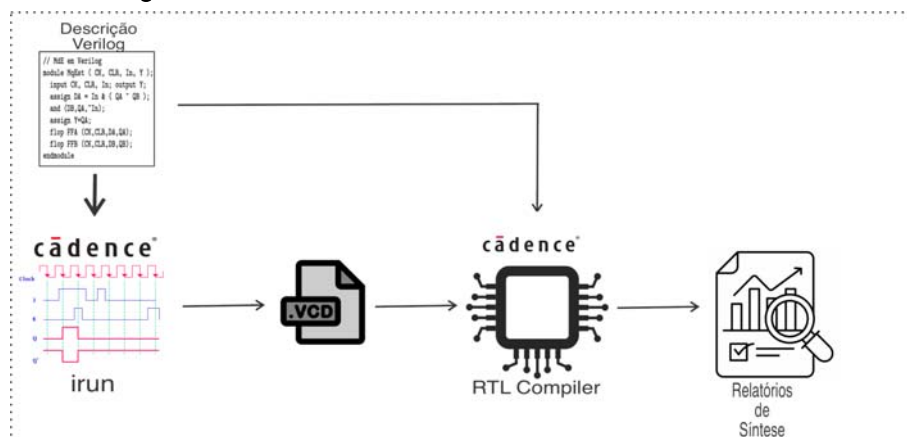
A arquitetura dos circuitos foi descrita usando a linguagem *System Verilog*, conhecida por sua robustez e eficácia na modelagem de sistemas eletrônicos complexos. O processo de síntese adotado está detalhadamente ilustrado na Figura 42, que apresenta as etapas e os fluxos utilizados.

Durante a síntese, os circuitos são submetidos a uma estimulação com 20% de atividade de chaveamento, permitindo uma análise do comportamento dinâmico dos componentes. Os códigos HDL processados pelo *RTL Compiler* resultam em relatórios detalhados, que incluem métricas cruciais como o tempo de propagação, consumo de energia e área ocupada por cada componente.

A metodologia *reg to reg* é adotada para a extração dos resultados finais. Essa abordagem é essencial para otimizar a propagação do sinal de um registrador para outro, fornecendo uma avaliação mais precisa do desempenho dos circuitos. Os resultados, com a dissipação de potência, serão apresentados em microwatts ( $\mu\text{W}$ ), e a área em nanômetros quadrados ( $\text{nm}^2$ ). Os relatórios gerados durante o processo de síntese serão explorados detalhadamente nesta análise. Uma observação a ser feita é que as entradas de *clock*, *reset* e *enable* foram omitidas nas Figuras e nas descrição

das interfaces, já que fazem parte das boas práticas no desenvolvimento de circuitos digitais. A próxima sessão detalhará cada módulo e os resultados de síntese.

Figura 42 – Metodologia de Síntese



## 6.2 Módulo Posições

O circuito é projetado para gerar coordenadas de início e fim para linhas e colunas, permitindo a divisão de um quadro de 128x128 em subquadros de 32x32. Essa capacidade é fundamental, conforme ilustrado na Figura 43. O módulo suporta métodos de busca como Preditor Mediano, Busca Inicial, Dois Pontos, Varredura (*Raster*) e Busca Refinada.

Além de gerar coordenadas para o módulo de corte (Figura 44), o circuito calcula a distância com base nas coordenadas do bloco de referência e determina o vetor de movimento. Novas coordenadas, distância e vetor de movimento são atualizados a cada pulso de *clock*.

Os módulos internos da arquitetura incluem:

- **Posições:** Responsável por armazenar o total de posições que o algoritmo de busca visitará. Esse módulo é essencial para limitar as buscas a pontos previamente estabelecidos;
- **Pontos Direcionais:** Armazena o conjunto de coordenadas a serem visitadas pelo algoritmo de busca. As coordenadas são armazenadas de forma concatenada. Por exemplo, a busca em diamante pequeno é registrada assim: 24'b111000000001001000000111. Nessa sequência, 111 000 corresponde a (-1,0), 000001 a (0,1), 001000 a (1,0) e 000111 a (0,-1). A cada conjunto de 3 bits, é gerada uma coordenada. Portanto, a cada 6 bits, obtemos um par de coordenadas X e Y. Essas coordenadas representam os pontos a serem investigados.

- **Algoritmo:** Indica a busca a ser executada, diferenciando, por exemplo, a busca Varredura (*Raster*) de buscas baseadas em coordenadas;
- **Mux Posições:** Estabelece o número de pontos a serem explorados pelo algoritmo, otimizando os recursos e restringindo as buscas;
- **Particionador de Bits:** Responsável pelo particionamento das posições enviadas pelo módulo Pontos Direcionais. Ele segmenta os bits do Mux Pontos Direcionais, enquanto o Módulo Contador de Posições restringe a busca às posições desejadas;
- **Contador de Posição/Coordenadas:** Gera as coordenadas para o módulo de corte com base no vetor de movimento. Também calcula a distância obtida pelo cálculo das novas coordenadas;
- **Varredura ou Raster:** Executa a busca *Raster* a partir do vetor de movimento encontrado nas etapas anteriores.

Figura 43 – Arquitetura do Posições

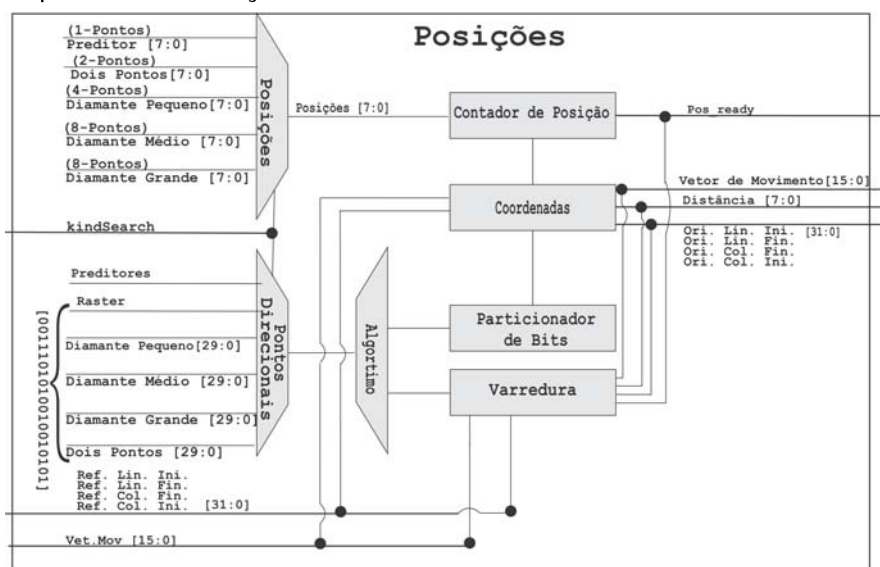


Tabela 32 – Interface Modulo Posição

| Porta         | Tipo    | Largura (bits) | Descrição                                             |
|---------------|---------|----------------|-------------------------------------------------------|
| KindSearch    | Entrada | 3              | Escolha do Algoritmo de Busca                         |
| Ref.Lin.XX    | Entrada | 8              | Coordenada do bloco de Referência                     |
| Ref.Col.XX    | Entrada | 8              | Coordenada do Bloco de Referência                     |
| Vet.Mov       | Entrada | 16             | Vetor de Movimento                                    |
| PosRedy       | Sáida   | 1              | Sinal de Pronto das Coordenadas                       |
| Vetor de Mov. | Sáida   | 16             | Vetor de Movimento da Coordenada Calculada            |
| Distânncia    | Sáida   | 16             | Distância da Coordenada Calculada em Relação a Origem |
| Ori.Lin.XX    | Sáida   | 8              | Coordenada Calculada do Bloco de Origem               |
| Ori.Col.XX    | Sáida   | 8              | Coordenada Calculada do Bloco de Origem               |

No desenvolvimento de circuitos digitais, a compreensão das interfaces é crucial. A Tabela 32, que detalha as interfaces, é fundamental para entender a dinâmica do módulo de posições. Este módulo possui quatro entradas e cinco saídas, cada uma designada com uma função específica e uma largura de bits determinada.

O processo de busca começa com a escolha de um algoritmo específico, marcado pela entrada *KindSearch*. Esta escolha se bifurca em duas trajetórias distintas: busca por coordenadas e busca *Varredura*. Ambas dependem do vetor de movimento fornecido na entrada para iniciar o processo.

No processo de busca por coordenadas, a primeira etapa envolve a segregação das coordenadas provenientes do módulo Pontos Direcionais. Uma vez segregadas, o cálculo das coordenadas é executado levando em consideração o ponto de segregação, o vetor de movimento e o contador de posição. O propósito dessa etapa é precisar a coordenada exata necessária para o procedimento de corte. Essa coordenada específica será posteriormente utilizada pelo módulo de Corte, juntamente com o vetor de movimento e a distância associada.

Paralelamente, e seguindo um trajeto distinto do dos Pontos Direcionais, inicia-se a busca por *Varredura* com o vetor de movimento inicial. A cada ciclo de clock, uma nova coordenada é calculada e enviada ao módulo de corte para uso imediato.

É importante salientar a diferença no funcionamento do contador de posições. Este contador registra o número de posições visitadas; por exemplo, na busca de dois pontos, como o próprio nome indica, dois pontos serão visitados. Esta configuração contrasta com a dos pontos diamante, que envolvem mais pontos.

Por fim, o módulo *Posições*, em sinergia com o TZS (*Test Zone Search*), é fundamental no processamento de imagens e codificação de vídeo. Ele não só fornece coordenadas precisas, mas também otimiza a comparação de blocos de imagens.

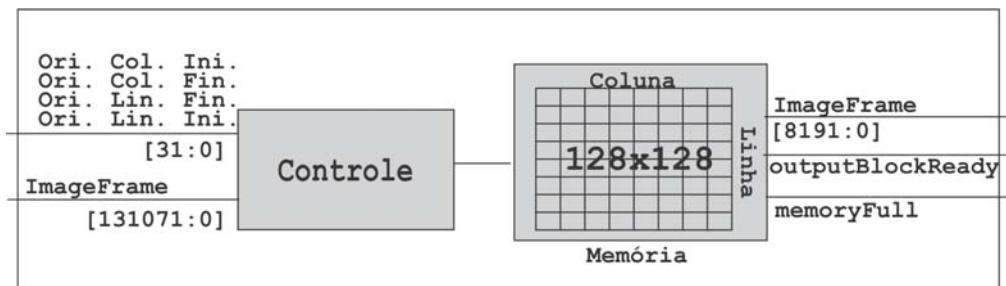
### 6.3 Módulo Corte

O Módulo Corte desempenha um papel fundamental na segmentação de quadros de imagem em sub-plocos candidatos 32x32 a partir de blocos maiores de 128x128, facilitando o cálculo da Diferença Absoluta de Soma (SAD). O processo se inicia com o carregamento dos quadros de referência na memória do sistema. Quando a memória atinge sua capacidade máxima, o sinal *memoryFull* é ativado, sinalizando que não é possível realizar mais armazenamentos. Este é o fim do ciclo operacional do módulo, que então se prepara para a fase seguinte do processamento de imagem. Segue-se a etapa de extração de um bloco específico, utilizando as coordenadas *Ori.Col.Ini*, *Ori.Col.Fin*, *Ori.Lin.Ini* e *EndCol* para definir a área de interesse. Uma vez que o bloco é extraído, ele é armazenado no registrador *outputBlock*, e o sinal *outputBlockReady* é acionado, indicando que o bloco está pronto para ser utilizado na próxima etapa,



que envolve o cálculo do SAD.

Figura 44 – Módulo de Corte



É importante salientar as interfaces do módulo, minuciosamente descritas na Tabela 33. Essa tabela oferece detalhes sobre cada entrada e saída do módulo, proporcionando uma compreensão clara de como os dados são gerenciados durante o processamento. O entendimento dessas informações é crucial não apenas para a apreciação da estrutura física do circuito, mas também para avaliar sua capacidade de processamento. Isso garante que o módulo opere eficientemente dentro dos parâmetros estabelecidos para a manipulação de vídeo. Além disso, a explicação detalhada de cada interface facilita a integração e otimização do módulo em sistemas mais amplos de processamento de imagem.

Tabela 33 – Interfaces Módulo de Corte

| Porta           | Tipo    | Largura | Descrição                               |
|-----------------|---------|---------|-----------------------------------------|
| Ori. Lin.XX     | Entrada | 3       | Esolha do algoritmo de busca            |
| Ori. Or.XX      | Entrada | 8       | Coordenada do Bloco de Referência       |
| ImageFrame      | Entrada | 131072  | Quadro para recorte                     |
| outputBlock     | Saída   | 9192    | Imagem extraída                         |
| utputBlockReady | Saída   | 1       | Indica que foi atualizado o outputblock |
| memoryFull      | Saída   | 1       | Indica que a memoria foi preenchida     |

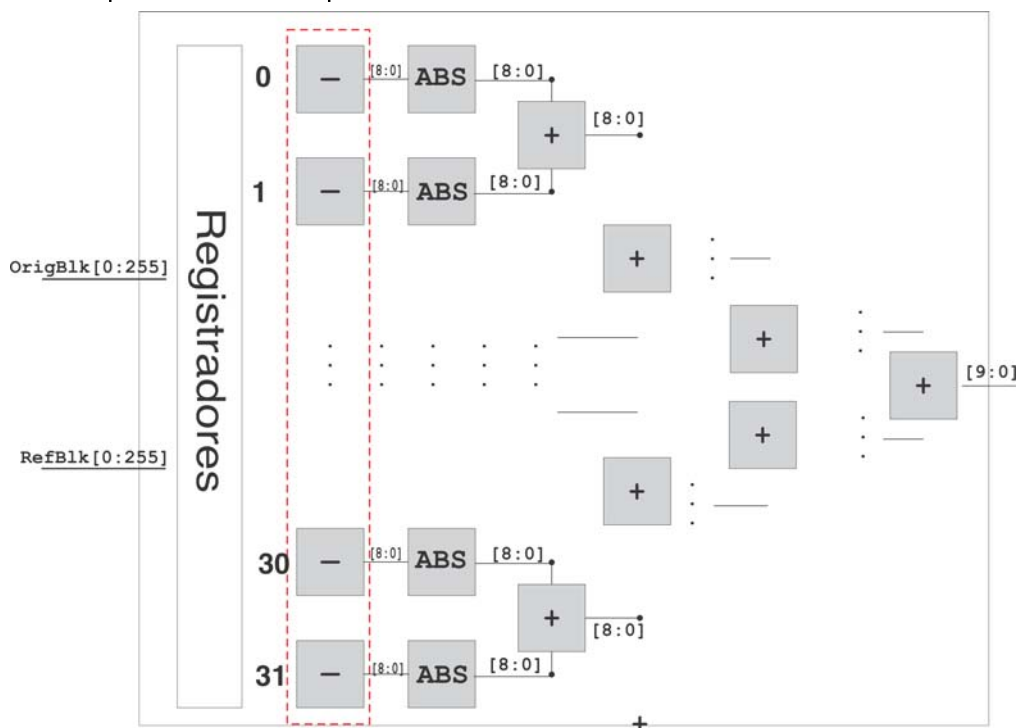
O Módulo Corte desempenha um papel crucial no funcionamento do algoritmo de busca TZS, sendo vital para a preparação dos dados necessários ao cálculo da Diferença Absoluta de Soma (SAD). Sua funcionalidade em segmentar os quadros de imagem em sub-plocos menores, além de gerenciar eficientemente o armazenamento e a extração de dados na memória, é fundamental para o aprimoramento do processamento subsequente. Devido a essa eficiência na manipulação de dados, o Módulo Corte se estabelece como um componente essencial na cadeia de processamento do TZS, contribuindo significativamente para a eficácia geral do algoritmo.

## 6.4 Módulo SAD

O módulo SAD, indispensável e objeto de estudo dessa tese, é projetado para calcular a Soma das Diferenças Absolutas em blocos de imagens de 32x32 pixels de 8 bits. Esse cálculo se desdobra em duas etapas principais: a subtração entre blocos

seguida pela soma das diferenças absolutas. Essa subtração pode ser efetuada por subtratores aproximados ou exatos, conforme a configuração.

Figura 45 – Arquitetura do SAD para blocos 32x32



As interfaces do módulo SAD são detalhadamente descritas na Tabela 34, que apresenta as portas de entrada e saída, cada uma com suas características específicas. Esta documentação é fundamental para o entendimento preciso da interação do módulo com outros componentes no sistema de processamento de imagens. A estru-

Tabela 34 – Interface SAD

| Porta    | Tipo    | Largura | Descrição                |
|----------|---------|---------|--------------------------|
| OriBkij  | Entrada | 32      | Bloco Original           |
| RefBkij  | Entrada | 32      | Bloco Referencia         |
| SAD      | Saída   | 10      | Resultado do SAD         |
| sadready | Saída   | 1       | Resultado foi atualizado |

tura de controle do módulo SAD garante uma operação eficiente, representando uma solução robusta para o cálculo da Soma das Diferenças Absolutas (SAD). Um aspecto distinto desta implementação é o uso de subtratores aproximados, uma inovação em relação às abordagens tradicionais documentadas na literatura.

Os resultados da síntese da arquitetura de SAD, que opera a uma frequência de 166 MHz, são expostos na Tabela 35. Esta síntese evidencia o desempenho dos subtratores aproximados em comparação ao *Full Subtractor*. A análise dos dados percentuais nas colunas da tabela permite uma comparação detalhada, indicando onde os subtratores aproximados superam o modelo exato. Valores negativos nessas colunas sinalizam uma diminuição efetiva em termos de número de *gates*, área

e potência dissipada, realçando a eficiência operacional destas variantes. As colunas com percentuais (%) apresentam a relação entre o subtrator aproximado e o subtrator *baseline*.

Quanto à simplificação de hardware, indicada pela redução no uso de portas lógicas (Gate(%)), os subtratores  $AXCS2_4$  e  $AXCS3_4$  mostram as melhores performances, ambos com uma diminuição de -2,87%. Essa redução no número de portas lógicas sugere uma simplificação substancial do hardware, que pode resultar em menores custos de produção e operação. Em contraste, o subtrator  $AppS_1$ , com uma redução mais modesta de -1,03%, revela uma eficiência menos expressiva.

Em relação à ocupação de área física (Area(%)), os subtratores  $AXCS2_4$  e  $AXCS3_4$  continuam a se destacar, com reduções de -2,24% e -2,10%, respectivamente. Essas diminuições não apenas poupam espaço físico, mas também contribuem para a eficiência geral dos sistemas nos quais são integrados, especialmente em dispositivos portáteis ou em sistemas onde o espaço é um recurso limitado. O  $AppS_1$  mostra uma vantagem mínima, com redução de apenas -0,03%, oferecendo pouca economia de espaço.

Na eficiência energética, a métrica Joules (%) mostra como os subtratores aproximados conseguem reduzir o consumo de energia em comparação ao subtrator completo. O modelo  $AXCS2_4$  se sobressai ao apresentar uma redução notável de -2,893% no consumo energético, estabelecendo-se como uma boa escolha para ambientes onde a eficiência energética é prioritária. Outros modelos como o  $AppS_4$  e o  $AXCS1_4$  também demonstram reduções consideráveis de -1,002% e -1,313%, respectivamente, evidenciando seu potencial para diminuir o consumo energético em sistemas digitais.

Os resultados desta análise confirmam que os subtratores aproximados, especialmente os modelos  $AXCS2_4$  e  $AXCS3_4$ , são opções eficientes para otimizar sistemas. Eles proporcionam reduções significativas no consumo de energia e na ocupação de área, sendo ideais para serem adotados em contextos tecnológicos onde a eficiência de recursos é crucial. Embora o subtrator  $AppS_1$  tenha algumas melhorias em comparação com o subtrator completo, seus benefícios são relativamente modestos, o que pode limitar sua aplicação em situações que exigem extrema economia de espaço ou orçamento. Essas informações são essenciais para designers e engenheiros que buscam componentes que maximizem a eficiência, minimizando o impacto ambiental e os custos operacionais. Em geral, os subtratores aproximados reduzem o número de portas lógicas (Gates), a área e se destacam pela eficiência energética.

Tabela 35 – Resultado Síntese do Módulo de SAD para frequência de 166 GHz e Potência em ( $\mu\text{W}$ )

| Arquitetura               | Gates  | Gate(%) | Area    | Area(%) | Linkage | Dynamic | Total   | Total(%) | Joules  | Joules(%) |
|---------------------------|--------|---------|---------|---------|---------|---------|---------|----------|---------|-----------|
| Full Subtractor           | 31.682 |         | 139.948 |         | 6,981   | 317,851 | 324,832 |          | 162,416 |           |
| <i>AppS</i> <sub>1</sub>  | 31.356 | -1,03   | 139.906 | -0,03   | 6,702   | 317,498 | 324,200 | -0,111   | 162,100 | -0,195    |
| <i>AppS</i> <sub>2</sub>  | 31.020 | -1,07   | 139.836 | -0,08   | 6,697   | 315,317 | 322,014 | -0,687   | 161,007 | -0,674    |
| <i>AppS</i> <sub>3</sub>  | 30.673 | -1,12   | 137.331 | -1,87   | 6,689   | 312,205 | 318,894 | -0,987   | 159,447 | -0,969    |
| <i>AppS</i> <sub>4</sub>  | 30.311 | -1,18   | 137.191 | -1,97   | 6,677   | 309,020 | 315,697 | -1,020   | 157,849 | -1,003    |
| <i>AXCS</i> <sub>11</sub> | 29.850 | -1,52   | 137.051 | -2,07   | 6,706   | 307,877 | 314,583 | -0,370   | 157,292 | -0,353    |
| <i>AXCS</i> <sub>12</sub> | 29.292 | -1,87   | 137.135 | -2,01   | 6,702   | 307,557 | 314,259 | -0,104   | 157,130 | -0,103    |
| <i>AXCS</i> <sub>13</sub> | 28.609 | -2,33   | 137.499 | -1,75   | 6,700   | 306,963 | 313,663 | -0,193   | 156,832 | -0,190    |
| <i>AXCS</i> <sub>14</sub> | 27.883 | -2,54   | 137.667 | -1,63   | 6,678   | 302,868 | 309,546 | -1,334   | 154,773 | -1,313    |
| <i>AXCS</i> <sub>21</sub> | 27.314 | -2,04   | 138.199 | -1,25   | 6,707   | 302,856 | 309,563 | -0,004   | 154,782 | 0,005     |
| <i>AXCS</i> <sub>22</sub> | 26.588 | -2,66   | 137.541 | -1,72   | 6,700   | 302,732 | 309,432 | -0,041   | 154,716 | -0,042    |
| <i>AXCS</i> <sub>23</sub> | 25.857 | -2,75   | 137.205 | -1,96   | 6,693   | 300,359 | 307,052 | -0,784   | 153,526 | -0,769    |
| <i>AXCS</i> <sub>24</sub> | 25.115 | -2,87   | 136.813 | -2,24   | 3,684   | 294,484 | 298,168 | -1,956   | 149,084 | -2,893    |
| <i>AXCS</i> <sub>31</sub> | 24.565 | -2,19   | 137.331 | -1,87   | 6,704   | 293,303 | 300,007 | -0,401   | 150,004 | 0,617     |
| <i>AXCS</i> <sub>32</sub> | 23.980 | -2,38   | 137.135 | -2,01   | 6,699   | 291,065 | 297,764 | -0,763   | 148,882 | -0,748    |
| <i>AXCS</i> <sub>33</sub> | 23.338 | -2,68   | 137.051 | -2,07   | 6,696   | 289,356 | 296,052 | -0,587   | 148,026 | -0,575    |
| <i>AXCS</i> <sub>34</sub> | 22.668 | -2,87   | 137.009 | -2,10   | 6,692   | 286,547 | 293,239 | -0,971   | 146,620 | -0,950    |

## 6.5 Módulo de Controle

O Módulo de controle, integrando uma Máquina de Estados Finitos (FSM), é responsável por controlar a busca por vetores de movimento no processamento de vídeo. A FSM, que opera de forma síncrona ao ciclo de relógio, é composta por 12 estados principais, cada um desempenhando uma função específica no processo de busca. Esta estrutura é baseada na Figura 10. A transição entre os estados é rigorosamente controlada por sinais e condições lógicas, que são determinadas pelos dados de entrada e pelas operações intermediárias. A configuração detalhada dessa máquina de estados é ilustrada na Figura 46. No Estado Inicial (INIT), a FSM prepara o módulo para um novo ciclo de busca, reiniciando todos os registros. Este estado permanece ativo até que as condições para o Carregamento (LOAD) sejam atendidas. No estado de Carregamento, o módulo carrega o bloco original e dá início ao processo de busca, prosseguindo para o estado Preditor (MED). Neste estado, realiza-se uma busca preditiva média, utilizando um vetor de movimento inicial, e atualizando os vetores de movimento temporários conforme necessário. A partir do estado INIT, todos os valores de SAD (Sum of Absolute Differences) resultantes são avaliados na busca pelo menor SAD.

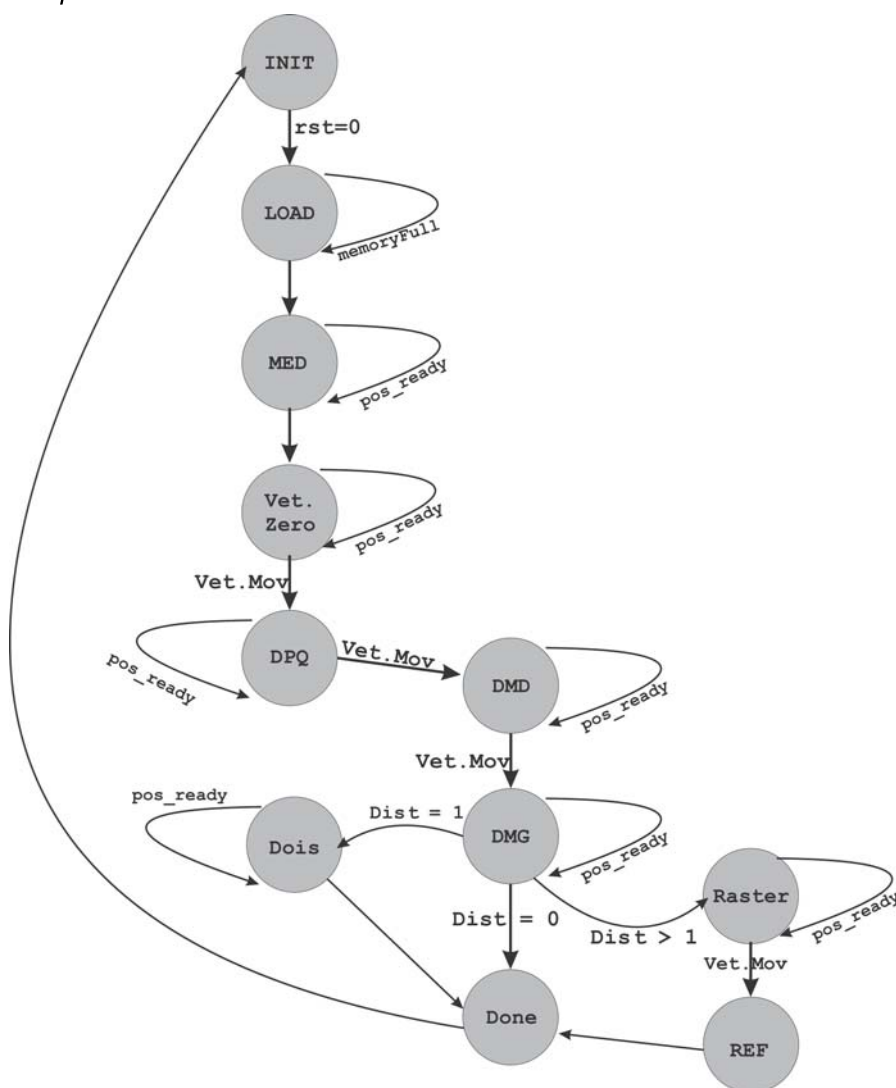
O estado subsequente, Preditor Zero, envolve o recálculo da SAD e o ajuste dos vetores de movimento. Em seguida, a FSM avança para a Busca de Ponto de Diamante Pequeno (DPQ), que realiza uma busca em escala reduzida. Mais informações sobre esta etapa estão disponíveis na Seção 2.3. Prosseguindo, ocorre a Busca de Diamante Médio (DMM), que expande o padrão de busca e explora posições mais afastadas, baseando-se sempre no vetor de movimento anterior. Esta fase também é descrita na Seção 2.3. Após a Busca de Diamante Médio, segue-se a Busca de Diamante Grande (DMG), que amplia ainda mais a área de busca. É importante mencionar que, neste caso, estamos implementando uma arquitetura conceitual do TZS e,

considerando que o bloco é de 32x32.

Dependendo do contexto, a FSM pode prosseguir para a Busca de Dois Pontos (DOIS) ou para a Varredura (*Raster*). Após completar a Varredura, a FSM transita para o estado de Referência (REF). Finalmente, a FSM chega ao estado Concluído (DONE), momento em que os resultados são registrados e o sistema é preparado para iniciar um novo ciclo de busca.

Assim, cada estado do módulo de TZS é projetado para identificar o vetor de movimento mais preciso, mantendo uma alta eficiência computacional. As transições entre os estados são estrategicamente controladas para maximizar a eficiência e a precisão, consolidando o módulo como uma ferramenta robusta e confiável para aplicações de compressão de vídeo em tempo real.

Figura 46 – Máquina de Estados TZS



## 6.6 Arquitetura TZS: Módulo Topo

A estrutura do módulo topo da arquitetura TZS, integrando todos os módulos discutidos anteriormente, desempenha um papel crucial no sistema. Essa arquitetura completa é ilustrada na Figura 41. O TZS não se limita a abranger os demais módulos, mas também coordena suas operações, conforme as interfaces descritas na Tabela 36.

Quando o módulo é ativado, inicia-se o processo no TZS com o carregamento de um bloco 128x128 no módulo de corte, uma etapa primordial para dar início às operações. Paralelamente, os registradores do módulo SAD são carregados com o bloco de referência, preparando o sistema para as tarefas que se seguem.

Tabela 36 – Interfaces Modulo TZS

| Porta             | Tipo    | Largura | Descrição                                      |
|-------------------|---------|---------|------------------------------------------------|
| Bloco Orig. 32x32 | Entrada | 256     | Entrada do bloco 32x32 para calculo do SAD     |
| Ref.Col e Ref.Lin | Entrada | 32      | Coordenadas do Bloco de Referência             |
| Preditor          | Entrada | 30      | Preditor Mediano                               |
| Bloco Ref.        | Entrada | 8192    | Bloco de Referência 32x32                      |
| SAD               | Saída   | 10      | Resultado do SAD                               |
| Distancia         | Saída   | 8       | Distancia bloco de Referência e Bloco original |

Seguindo essa preparação, o TZS processa as coordenadas do bloco de referência. Nesse ponto, o vetor de movimento é introduzido no módulo de posições, responsável por calcular a coordenada, o vetor de deslocamento e a distância do vetor. Essa fase é crítica, estabelecendo os parâmetros para as operações subsequentes.

Concluída essa etapa, o módulo de corte é ativado, já recebendo as coordenadas do módulo de posições. Ele então extrai sub-plocos de sua memória, um processo essencial para que os sub-plocos sejam posteriormente utilizados no cálculo do SAD. Essa sequência de operações sublinha a eficiência e interconexão dos módulos no sistema TZS, realçando a complexidade do design.

A eficácia do módulo TZS torna-se ainda mais evidente ao analisarmos sua performance. Este foi sintetizado para operar a uma frequência de 142 MHz, uma característica que contribui significativamente para sua alta capacidade de processamento. Em uma análise do pior cenário, onde todas as buscas são necessárias, o circuito demanda aproximadamente 890 ciclos de clock para concluir uma resposta. Notavelmente, uma grande parcela desse tempo, especificamente 841 iterações, é dedicada à busca de Varredura.

O *throughput* desse circuito TZS é de aproximadamente 159.550 buscas por segundo. Isso indica que, mesmo sob condições extremas, o circuito é capaz de processar cerca de 159.550 blocos de 32x32 em uma área de busca de 128x128 cada segundo. Essa capacidade demonstra a eficiência e a rapidez com que o módulo TZS pode tratar informações visuais.

A análise da Tabela 37 detalhada dos resultados referentes às reduções e aumentos das colunas de Gates, Área e Total. Na categoria de Gates, a arquitetura *AXCS2<sub>4</sub>*

apresentou a melhor performance com uma redução de -4,865%. Este resultado indica uma melhoria que pode conduzir a menores custos de fabricação e redução no consumo de energia. Em contrapartida, a arquitetura  $AppS_2$  exibiu a menor redução, com apenas -0,445%, sugerindo limitações para aplicações onde o espaço ou custo são fatores críticos. Em resação à Área, o melhor desempenho foi observado na arquitetura  $AXCS3_4$ , com uma redução de -0,49%. Embora modesta, esta é a redução mais significativa entre as arquiteturas estudadas e pode facilitar um aumento na densidade de componentes nos chips. Por outro lado, a arquitetura  $AXCS3_2$  mostrou um aumento na área ocupada de 0,52%, o que pode refletir um trade-off onde melhorias em outros aspectos levaram a um design menos compacto.

Quanto à Redução Total, a arquitetura  $AppS_4$  se destacou com uma redução de -1,807%, demonstrando a maior eficiência global ao equilibrar várias métricas de desempenho e custo. Em contraste, a arquitetura  $AXCS2_1$  apresentou um aumento total de 1,05%, o que pode indicar desvantagens em sua aplicação onde a eficiência é prioritária, possivelmente priorizando outros atributos como precisão ou velocidade em detrimento da economia de recursos.

Tabela 37 – Resultados de Potência ( $\mu W$ ) Frequência para frequência de 142 Mhz

| Arquivo         | gates   | Gate%  | Total   | Area% | Leakage | Dynamic | Total  | Total% |
|-----------------|---------|--------|---------|-------|---------|---------|--------|--------|
| Full Subtractor | 210.477 |        | 858.413 |       | 27,87   | 763,55  | 791,42 |        |
| $AppS_1$        | 209.453 | -0,487 | 857.933 | -0,06 | 28,96   | 762,70  | 791,66 | -0,111 |
| $AppS_2$        | 209.541 | -0,445 | 857.923 | -0,06 | 27,93   | 762,48  | 790,40 | -0,141 |
| $AppS_3$        | 206.381 | -1,946 | 857.692 | -0,08 | 27,96   | 749,98  | 777,93 | -1,778 |
| $AppS_4$        | 202.285 | -3,892 | 856.006 | -0,20 | 27,96   | 749,75  | 777,71 | -1,807 |
| $AXCS1_1$       | 209.452 | -0,487 | 858.413 | 0,28  | 27,87   | 760,88  | 788,75 | -0,350 |
| $AXCS1_2$       | 209.453 | -0,487 | 858.172 | -0,03 | 27,98   | 756,07  | 784,05 | -0,980 |
| $AXCS1_3$       | 208.429 | -0,973 | 857.933 | -0,03 | 28,09   | 755,76  | 783,85 | -1,020 |
| $AXCS1_4$       | 207.405 | -1,460 | 859.619 | 0,20  | 28,27   | 753,37  | 781,64 | -1,334 |
| $AXCS2_1$       | 208.429 | -0,973 | 857.690 | -0,22 | 27,50   | 771,57  | 799,07 | 1,050  |
| $AXCS2_2$       | 206.381 | -1,946 | 853.598 | -0,48 | 27,69   | 759,21  | 786,90 | -0,569 |
| $AXCS2_3$       | 202.285 | -3,892 | 855.282 | 0,20  | 27,81   | 756,07  | 783,88 | -0,980 |
| $AXCS2_4$       | 200.237 | -4,865 | 854.078 | -0,14 | 27,76   | 753,98  | 781,74 | -1,254 |
| $AXCS3_1$       | 209.421 | -0,502 | 854.001 | -0,01 | 28,65   | 758,67  | 787,32 | -0,640 |
| $AXCS3_2$       | 208.425 | -0,975 | 858.410 | 0,52  | 27,89   | 756,72  | 784,61 | -0,895 |
| $AXCS3_3$       | 208.125 | -1,117 | 858.310 | -0,01 | 27,65   | 755,76  | 783,41 | -1,020 |
| $AXCS3_4$       | 208.025 | -1,165 | 854.078 | -0,49 | 27,13   | 755,53  | 782,66 | -1,050 |

## 6.7 Discussão dos Resultados

A Tabela 38 apresenta um resumo detalhado dos resultados obtidos nesta tese, incluindo os aspectos já discutidos na Tabela 31. A QtdSAD acumula a quantidade de SADS executados incluindo os do apêndice B Análise para Blocos de Tamanhos fixos. O BD-RATE acumula a média geral incluindo os dados do mesmo apêndice e dissipação de potência e energia são os dados discutidos anteriormente. A tabela apresenta uma visão geral dos resultados obtidos em toda a tese.

Essas informações nos permitem obter uma compreensão bastante precisa da realidade estudada. Ao avaliar o desempenho de subtratores aproximados em arquiteturas de codificação de vídeo, inicialmente se observa que esses subtratores mantêm



a quantidade de SADs (Sum of Absolute Differences) muito próxima à do subtrator exato, com variações percentuais mínimas. Esta observação indica uma eficiência computacional equivalente, o que facilita a adoção desses subtratores em ambientes onde o desempenho é crítico, como em dispositivos móveis e sistemas embarcados, sem comprometer a autonomia da bateria onde a otimização de energia e recursos é essencial.

Além disso, a eficiência de codificação, medida pelo índice BD-RATE, é minimamente impactada. A maioria dos subtratores aproximados apresenta pequenas variações em BD-RATE, indicando uma perda insignificante de qualidade, o que reforça a viabilidade de sua utilização sem comprometer significativamente a integridade do conteúdo visual. Ao examinar os resultados da tabela de eficiência de codificação e consumo energético de várias arquiteturas de subtratores aproximados, torna-se evidente que algumas arquiteturas se destacam tanto pelos avanços quanto pelos desafios que apresentam. Com foco nos aspectos de redução de consumo de energia e potência, bem como na eficiência das operações de Soma das Diferenças Absolutas (SAD), identificamos arquiteturas que demonstram otimizações significativas e outras que indicam potencial para melhorias.

Os subtratores  $AppS_4$  e  $AppS_3$  emergem como exemplos de redução. O  $AppS_4$ , em particular, apresenta uma redução de -1,807% na dissipação de potência e de -1,733% no consumo energético, apesar de um aumento leve de 0,43% nas operações de SAD. Este subtrator exemplifica como ajustes na arquitetura podem resultar em ganhos substanciais em termos de sustentabilidade energética sem comprometer significativamente o desempenho. O  $AppS_3$  também se mostra eficaz, com reduções de -1,778% em potência e -1,705% em energia, acompanhadas de um aumento modesto de 0,099% em operações de SAD. Estes resultados sublinham a capacidade dessa arquitetura de equilibrar precisão e eficiência energética.

Por outro lado, o  $AXCS2_1$  representa um caso que necessita de atenção devido ao seu aumento de 1,050% no consumo de potência e de 0,967% no consumo energético, apesar do aumento de apenas 0,805% nas operações de SAD. Este subtrator ilustra os desafios associados a algumas arquiteturas que, embora incrementem ligeiramente o número de operações processadas, fazem-no ao custo de um consumo energético substancialmente maior.

Os subtratores aproximados como  $AppS_4$  e  $AppS_3$  são vitrines de como inovações em design de circuitos podem contribuir significativamente para a redução do consumo energético, mantendo a funcionalidade necessária para aplicações exigentes. A eficiência demonstrada por essas arquiteturas não só apoia a adoção de subtratores aproximados como uma estratégia viável para melhorar a sustentabilidade energética, mas também estabelece um padrão que outras arquiteturas devem aspirar a alcançar. Enquanto isso, os desafios observados no  $AXCS2_1$  ressaltam a importância de



um desenvolvimento contínuo e refinado para maximizar tanto a eficiência energética quanto o desempenho de processamento.

Essas análises não apenas validam a utilidade dos subtratores aproximados em contextos práticos, mas também fornecem diretrizes claras para futuras pesquisas e desenvolvimento na área de codificação de vídeo e computação aproximada, com o objetivo de harmonizar eficiência energética e precisão computacional.

Tabela 38 – Desempenho dos Subtratores na Contagem de SADs Executados e Resultado de Potência ( $\mu\text{W}$ ) e Energia(nJ)

| Subtractor                | QtdSAD         | QtdSAD(%) | BD-RATE | Potência | Potência(%) | Joules | Joules(%) |
|---------------------------|----------------|-----------|---------|----------|-------------|--------|-----------|
| Full                      | 25.726.762.029 |           |         | 791,42   |             | 1,188  |           |
| <i>AppS</i> <sub>1</sub>  | 25.726.762.029 | 0         | 0       | 791,66   | -0,111      | 1,189  | 0,030     |
| <i>AppS</i> <sub>2</sub>  | 25.744.761.620 | 0,07      | -0,047  | 790,40   | -0,141      | 1,187  | -0,129    |
| <i>AppS</i> <sub>3</sub>  | 25.752.309.053 | 0,099     | 0,011   | 777,93   | -1,778      | 1,168  | -1,705    |
| <i>AppS</i> <sub>4</sub>  | 25.837.338.984 | 0,43      | -0,004  | 777,71   | -1,807      | 1,168  | -1,733    |
| <i>AXCS</i> <sub>11</sub> | 25.780.715.696 | 0,21      | -0,015  | 788,75   | -0,350      | 1,184  | -0,338    |
| <i>AXCS</i> <sub>12</sub> | 25.731.738.721 | 0,019     | 0,025   | 784,05   | -0,980      | 1,177  | -0,932    |
| <i>AXCS</i> <sub>13</sub> | 25.949.774.237 | 0,867     | 0,012   | 783,85   | -1,020      | 1,177  | -0,956    |
| <i>AXCS</i> <sub>14</sub> | 25.838.139.287 | 0,433     | 0,006   | 781,64   | -1,334      | 1,174  | -1,236    |
| <i>AXCS</i> <sub>21</sub> | 25.933.781.865 | 0,805     | 0,029   | 799,07   | 1,050       | 1,200  | 0,967     |
| <i>AXCS</i> <sub>22</sub> | 25.859.231.366 | 0,515     | -0,008  | 786,90   | -0,569      | 1,182  | -0,571    |
| <i>AXCS</i> <sub>23</sub> | 26.024.164.268 | 1,156     | -0,005  | 783,88   | -0,980      | 1,177  | -0,953    |
| <i>AXCS</i> <sub>24</sub> | 26.066.673.975 | 1,321     | 0,06    | 781,74   | -1,254      | 1,174  | -1,224    |
| <i>AXCS</i> <sub>31</sub> | 26.130.990.617 | 1,571     | 0,029   | 787,32   | -0,640      | 1,182  | -0,519    |
| <i>AXCS</i> <sub>32</sub> | 26.101.657.113 | 1,457     | 0,009   | 784,61   | -0,895      | 1,178  | -0,861    |
| <i>AXCS</i> <sub>33</sub> | 26.461.497.871 | 2,856     | -0,046  | 783,41   | -1,020      | 1,176  | -1,012    |
| <i>AXCS</i> <sub>34</sub> | 26.201.518.637 | 1,845     | -0,047  | 782,66   | -1,050      | 1,175  | -1,107    |

## 6.8 Discussão Frente aos Trabalhos Relacionados

No campo de codificação de vídeo, estudos recentes têm se concentrado em melhorar a eficiência energética e a precisão de cálculos envolvendo o SAD. A contribuição de El Harouni *et al.* (El-harouni et al., 2017) é notável nesse contexto, onde implementaram somadores aproximados na árvore de somadores do SAD. Eles exploraram variações de precisão de 1 a 6 bits, desenvolvendo aceleradores SAD que combinam somadores precisos e aproximados, com escolhas variando entre 2, 4 ou 6 LSBs para aproximação. Esses somadores foram integrados em unidades SAD de 8x8 e 32x32 no fluxo de design ASIC para codificação interquadros. Em uma abordagem complementar, Porto *et al.* (Porto et al., 2017, 2019) investigaram a estimativa de movimento no cálculo do SAD, utilizando somadores *Lower-Part-OR* aproximados (LOA). Esta técnica visa reduzir o consumo de energia mantendo a eficiência de codificação. Eles desenvolveram três arquiteturas VBSME (*Variable Block Size Motion Estimation*), que incorporam operadores LOA com diferentes níveis de imprecisão. Avançando nesta linha de pesquisa, Lima *et al.* (Lima et al., 2021) aplicaram a computação aproximada em somadores e subtratores na codificação interquadros. Seu método envolveu a remoção de colunas de operadores aritméticos da Transformada de Hadamard 8x8, gerando versões aproximadas do SATD que contribuem para a eficiência energética, sem comprometer a compressão. Seguindo uma direção similar,

Paltrinieri *et al.* (Paltrinieri et al., 2019) exploraram a substituição de somadores precisos por versões aproximadas nas métricas SATD e SAD, equilibrando a economia de energia e a qualidade do vídeo.

O estudo de Abreu *et al.* (Abreu et al., 2017) focou no uso de um operador absoluto aproximado em arquiteturas SAD. Eles compararam diferentes arquiteturas, analisando a redução do consumo de energia, área do circuito e dissipação de potência. Em uma pesquisa relacionada, Ahmad *et al.* (Ahmad; Ayrancioglu; Hamzaoglu, 2020) se concentraram em operadores absolutos em arquiteturas SAD, visando otimizar o consumo de energia e a área do circuito através de um novo somador proposto. Por fim, o trabalho Chong *et al.* (Chong; Ortega, 2007) analisou a aplicação de soluções baseadas na modelagem de erros de computação devido à escala de tensão e subamostragem do SAD.

Finalmente, relembra-se que estudos abordando subtratores aproximados são pouco explorados na literatura atual. Essa observação é decisiva, especialmente considerando a frequente utilização de somadores aproximados no cálculo de SAD. Esta tese buscou cobrir esta lacuna significativa nos estudos sobre o impacto real dessa aproximação nos algoritmos de busca do TZS. Além disso, análises completas de arquiteturas de TZS que integram esses somadores aproximados são raras. Este estudo contribui para preencher essa lacuna, explorando o potencial dos subtratores em sistemas de codificação de vídeo e otimização de consumo energético destacando a importância de uma abordagem mais abrangente na análise de arquiteturas de codificação de vídeo. Os autores desconhecem outros trabalhos que propõe novos subtratores aproximados e avaliam seu impacto, em área, energia e qualidade (eficiência de codificação), quando aplicados a arquiteturas de hardware dedicadas à predição interquadros em codificadores de vídeo atuais como o VVC.

## 7 CONCLUSÃO

Nesta conclusão, destacamos o sucesso na realização dos objetivos propostos e na resposta às questões de pesquisa. O objetivo principal de aumentar a eficiência energética nos sistemas de codificação de vídeo foi alcançado com a implementação dos subtratores aproximados. Demonstrou-se que esses subtratores são capazes de equilibrar eficiência energética e precisão computacional, apresentando uma solução otimizada para a codificação de vídeo.

*Questão de Pesquisa 1 (Q1): É possível desenvolver subtratores aproximados energeticamente eficientes para aplicação na codificação de vídeo do CODEC VVC?* Verificou-se que os subtratores aproximados desenvolvidos contribuem significativamente para a eficiência energética, sem comprometer a precisão necessária aos processos de codificação de vídeo.

*Questão de Pesquisa 2 (Q2): A operação de Soma das Diferenças Absolutas (SAD), utilizada nas etapas de predição, tolera bem as aproximações no codificador VVC?* Os experimentos confirmaram que a operação de Soma das Diferenças Absolutas (SAD) é tolerante à aproximação, possibilitando uma implementação que favorece tanto a precisão quanto a eficiência energética.

*Questão de Pesquisa 3 (Q3): É viável desenvolver um sistema de hardware dedicado para predição interquadros no codec VVC que utilize subtratores aproximados nas operações de SAD?* Foi respondida com a viabilidade de projetar um sistema de hardware dedicado para predição interquadros no padrão VVC, incorporando os subtratores aproximados. Esse desenvolvimento marca um avanço significativo no padrão VVC, ressaltando o impacto prático da pesquisa.

Este trabalho destaca-se não só pela sua contribuição, mas também pela inovação ao aplicar, pela primeira vez, subtratores aproximados na árvore de SAD. Este desenvolvimento, juntamente com a melhoria das métricas nas etapas do algoritmo TZS e a criação de uma nova arquitetura para estimativa de movimento, marcam um avanço

considerável no campo. Essas inovações reforçam tanto a originalidade quanto a relevância prática da pesquisa realizada.

A adoção de subtratores aproximados representa uma estratégia promissora para otimizar o consumo de energia em processos computacionais principalmente na codificação de vídeos digitais. Essa abordagem se destaca especialmente em aplicações onde a eficiência energética é crucial, sem comprometer significativamente a precisão dos cálculos. A eficácia dos subtratores aproximados é evidenciada ao analisar as métricas de desempenho relacionadas à potência e ao consumo energético, bem como a quantidade de operações de SAD realizadas.

Os subtratores aproximados, como  $AppS_1$  e  $AppS_4$ , demonstram capacidades distintas de redução de consumo que podem ser decisivas dependendo dos requisitos específicos de cada aplicação. O subtrator  $AppS_1$ , por exemplo, mostra uma redução na potência consumida sem alterar o número de operações de SAD. Essa característica indica uma melhoria modesta, mas significativa, caso a conservação de energia seja uma prioridade sem a necessidade de incrementar a carga de processamento.

Por outro lado, o subtrator  $AppS_4$  se sobressai devido à sua capacidade de reduzir significativamente tanto a potência quanto o consumo energético, com apenas um leve aumento nas operações de SAD. Esse perfil sugere um bom equilíbrio entre a economia de energia e a melhoria do processamento, adequado para cenários em que pequenos ganhos em precisão justificam grandes economias energéticas. A adoção de  $AppS_4$  pode ser particularmente vantajosa em ambientes que exigem extensos cálculos de SAD, como em processamento de vídeo e tarefas de visão computacional.

Além dessas vantagens operacionais, é importante destacar que, mesmo com o cálculo aproximado de SAD realizado por esses subtratores, os valores de BDRate — uma métrica que avalia a eficiência de codificação — mostram quase nenhuma alteração. Isso indica que a adoção de subtratores aproximados não compromete a qualidade do resultado final, mantendo a eficácia na codificação de vídeos ou imagens, que é fundamental em muitas aplicações modernas de tecnologia.

Em conclusão, os subtratores aproximados representam uma alternativa vantajosa do ponto de vista energético para a execução de operações de SAD, mantendo um impacto praticamente nulo sobre a qualidade da codificação. Esses achados demonstram a viabilidade dos subtratores aproximados como uma opção estratégica para projetos que buscam não apenas a otimização de recursos, mas também a promoção da sustentabilidade energética.

Ademais, esta pesquisa traz uma contribuição inovadora para o domínio da computação aproximada, estabelecendo avanços na área de codificação de vídeo e delineando diretrizes para futuros estudos. Assim, este estudo não apenas alcança seus objetivos iniciais e confirma suas hipóteses, mas também representa um avanço tecnológico significativo no campo da codificação de vídeo. Ele exemplifica a combinação

de eficiência energética com precisão computacional, estabelecendo um modelo para futuros avanços na área.

## 7.1 Trabalhos Futuros

Para futuros trabalhos, sugere-se um estudo aprofundado sobre a influência das variações dos subtratores aproximados nas distintas fases do algoritmo *Test Zone Search* (TZS). Em nossa investigação, limitamo-nos a avaliar um único operador por etapa no TZS. Entretanto, análises complementares poderiam se beneficiar da exploração de operadores distintos em cada fase, enriquecendo o entendimento acerca do impacto dessas variações na eficácia e na precisão do procedimento.

Um outro enfoque relevante para investigações futuras é a arquitetura do TZS. A configuração atual é restrita a posições inteiras, desprovida de suporte para interpolação fracionária - uma limitação que, se superada, poderia representar um avanço significativo. Além disso, a arquitetura presente contempla apenas blocos de dimensões fixas, excluindo a possibilidade de adaptar-se a blocos de tamanhos variáveis ou de formatos retangulares. A expansão dessa flexibilidade poderia resultar em melhorias na adaptabilidade e na eficiência do sistema de codificação de vídeo. Existe, ainda, o potencial de integrar subtratores, somadores e operadores de valor absoluto aproximados de maneira conjunta.

Essas proposições de avanço detêm a capacidade de contribuir significativamente para o desenvolvimento da codificação de vídeo, promovendo um equilíbrio entre eficiência energética e precisão computacional. Tais inovações pavimentam o caminho para o surgimento de novos conhecimentos e aplicações práticas na esfera da codificação de vídeo.

## REFERÊNCIAS

ABREU, B.; PAIM, G.; GRELLERT, M.; SILVEIRA, B.; DINIZ, C.; COSTA, E.; BAMPI, S. Exploiting absolute arithmetic for power-efficient sum of absolute differences. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS (ICECS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.522–525.

AGOSTINI, L. V. Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas à Compressão de Vídeo Segundo o Padrão H.264/AVC. , [S.l.], p.172, 2007.

AHMAD, W.; AYRANCIOGLU, B.; HAMZAOGLU, I. Comparison of Approximate Circuits for H.264 and HEVC Motion Estimation. In: EUROMICRO CONFERENCE ON DIGITAL SYSTEM DESIGN (DSD), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.167–173.

AL., F. et. Fractional Motion Estimation in VVC. **IEEE Xplore**, [S.l.], 2024. Fractional Motion Estimation (FME) with six-point Gaussian filter.

AL., W. et. Adaptive Motion Vector Resolution in VVC. **IEEE Xplore**, [S.l.], 2024. Adaptive Motion Vector Resolution (AMVR) for precise motion vector coding.

AL., Y. et. Low-Complexity VVC Optimization. **arXiv**, [S.l.], 2024. Angular intra-frame prediction modes for improved efficiency.

ANDREWS, J. G.; BUZZI, S.; CHOI, W.; HANLY, S. V.; LOZANO, A.; SOONG, A. C. K.; ZHANG, J. C. What Will 5G Be? **IEEE Journal on Selected Areas in Communications**, [S.l.], v.32, n.6, p.1065–1082, 2014.

BARUA, H. B.; MONDAL, K. C. Approximate Computing: A Survey of Recent Trends—Bringing Greenness to Computing and Communication. **Journal of The Institution of Engineers (India): Series B**, [S.l.], 2019.

BOSSEN, F. Common test conditions and software configurations. In: JCT-VC DOCUMENT NO. L1100, 2013. **Anais...** [S.l.: s.n.], 2013.

BOSSEN, F.; BROSS, B.; SUHRING, K.; FLYNN, D. HEVC Complexity and Implementation Analysis. **IEEE Trans. Circuits Syst. Video Technol.**, [S.l.], v.22, n.12, p.1685–1696, Dec 2012.

BROSS, B.; ANDERSSON, K.; BLASER, M.; DRUGEON, V.; KIM, S. H.; LAINEMA, J.; LI, J.; LIU, S.; OHM, J. R.; SULLIVAN, G. J.; YU, R. General Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression with Capability beyond HEVC. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.30, n.5, p.1226–1240, 2020.

BROSS, B.; WANG, Y. K.; YE, Y.; LIU, S.; CHEN, J.; SULLIVAN, G. J.; OHM, J. R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, p.3736–3764, 10 2021.

CASSA, M. B.; NACCARI, M.; PEREIRA, F. Fast rate distortion optimization for the emerging HEVC standard. In: PICTURE CODING SYMPOSIUM, 2012., 2012. **Anais...** [S.l.: s.n.], 2012. p.493–496.

CHEN, J.; YE, Y.; KIM, S. H. **Test Model 8 for Versatile Video Coding (VTM 8)**. [S.l.]: Joint Video Experts Team, 2020.

CHEN, L.; HAN, J.; LIU, W.; LOMBARDI, F. Design of approximate unsigned integer non-restoring divider for inexact computing. **Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI**, [S.l.], v.20-22-May-, n.c, p.51–56, 2015.

CHIPPA, V. K.; CHAKRADHAR, S. T.; ROY, K.; RAGHUNATHAN, A. Analysis and characterization of inherent application resilience for approximate computing. **Proceedings - Design Automation Conference**, [S.l.], n.i, 2013.

CHONG, I. S.; ORTEGA, A. Power efficient motion estimation using multiple imprecise metric computations. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO, 2007., 2007. **Anais...** [S.l.: s.n.], 2007. p.2046–2049.

CHRISTIAN fraunhofer. **VVC: Open-GOP Resolution Switching**. Disponível em: <<https://bitmovin.com/vvc-open-gop-resolution-switching>>. Acesso em: 2023-09-09.

CLEMENT, J. **Worldwide digital population as of July 2019**. Disponível em: <<https://www.statista.com/statistics/617136/digital-population-worldwide/>>. Acesso em: 10-28-2019.

CORREA, G. R. **Computational Complexity Reduction and Scaling for High Efficiency Video Encoders**. 2014. Tese (Doutorado em Ciência da Computação) — Universidade de Coimbra.

Davidson, S. Good Enough Computing. **IEEE Design Test**, [S.l.], v.33, n.1, p.96–96, Feb 2016.

De-Luxán-Hernández, S.; George, V.; Ma, J.; Nguyen, T.; Schwarz, H.; Marpe, D.; Wiegand, T. An Intra Subpartition Coding Mode for VVC. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1203–1207.

DOAN, N.; KIM, T. S.; RHEE, C. E.; LEE, H.-J. A hardware-oriented concurrent TZ search algorithm for High-Efficiency Video Coding. **EURASIP journal on advances in signal processing**, [S.l.], v.2017, n.1, p.1–17, 2017.

DOU, Y.; WANG, C.; WOODS, R.; LIU, W. ENAP: An Efficient Number-Aware Pruning Framework for Design Space Exploration of Approximate Configurations. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v.70, n.5, p.2062–2073, 2023.

EL-HAROUNI, W.; REHMAN, S.; PRABAKARAN, B. S.; KUMAR, A.; HAFIZ, R.; SHAFIQUE, M. Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding. **Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017**, [S.l.], p.1384–1389, 2017.

FATEMIEH, S. E.; RESHADINEZHAD, M. R.; TAHERINEJAD, N. Fast and Compact Serial IMPLY-Based Approximate Full Adders Applied in Image Processing. **IEEE Journal on Emerging and Selected Topics in Circuits and Systems**, [S.l.], v.13, n.1, p.175–188, 2023.

FELDMANN. **Versatile Video Coding (VVC) JVET**. Disponível em: <[https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_vTM](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_vTM)> .*Acesso em* : 2021 – 01 – 12.

GAO, H.; ESENLIK, S.; ALSHINA, E.; STEINBACH, E. Geometric Partitioning Mode in Versatile Video Coding: Algorithm Review and Analysis. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], 2020.

GOGOI, S.; PEESAPATI, R. A motion estimation search algorithm and its hardware implementation for HEVC/H. 265. In: IEEE 10TH INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS (ICCE-BERLIN), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–6.

GONCALVES, P.; CORREA, G.; AGOSTINI, L.; PORTO, M. Learning-based bypass zone search algorithm for fast motion estimation. **Multimedia Tools and Applications**, [S.l.], v.82, n.3, p.3535–3560, 2023.



GONCALVES, P.; MORAES, C.; CORREA, G.; PORTO, M. Complexity-aware TZS algorithm for high efficiency video coding. **Journal of Integrated Circuits and Systems**, [S.l.], v.14, p.1–9, 2019.

GUPTA, V.; MOHAPATRA, D.; PARK, S. P.; RAGHUNATHAN, A.; ROY, K. IMPACT: IMPrecise adders for low-power approximate computing. **Proceedings of the International Symposium on Low Power Electronics and Design**, [S.l.], p.409–414, 2011.

GUPTA, V.; MOHAPATRA, D.; RAGHUNATHAN, A.; ROY, K. Low-power digital signal processing using approximate adders. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.l.], v.32, n.1, p.124–137, 2013.

HAN, J.; ORSHANSKY, M. Approximate computing: An emerging paradigm for energy-efficient design. In: IEEE EUROPEAN TEST SYMPOSIUM (ETS), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.1–6.

HENNESSY, J.; PATTERSON, D. **Computer Architecture: A Quantitative Approach**. 6.ed. Waltham, Mass.: Morgan Kaufmann, 2017.

HEVC Test Model (HM) v. 16.9. <http://HEVC.hhi.fraunhofer.de>.

HHI, F. **VVC Overview**. Accessed: June 14, 2024. Disponível em: <<https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/video-compression-and-analysis/vvc.html>>.

HIVO. **Maximizing Storage: The Benefits of Video Compression**. Accessed: June 14, 2024. Disponível em: <<https://www.hivo.co/blog/maximizing-storage-the-benefits-of-video-compression>>.

IBGE. **Informações Atualizadas sobre Tecnologias da Informação e Comunicação**. Acessado em 01 de Janeiro de 2024, <https://educa.ibge.gov.br/jovens/materias-especiais/21581-informacoes-atualizadas-sobre-tecnologias-da-informacao-e-comunicacao.html>.

ITAIPU. **Produção ano a ano | ITAIPU BINACIONAL**. Disponível em: <<https://www.itaipu.gov.br/energia/producao-ano-ano>>.

ITU. **New 'Versatile Video Coding' standard to enable next-generation video compression**. Accessed: June 14, 2024. Disponível em: <<https://www.itu.int/en/mediacentre/Pages/2020-PR03.aspx>>.

JIANG, H.; HAN, J.; LOMBARDI, F. A comparative review and evaluation of approximate adders. **Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI**, [S.l.], v.20-22-May-, n.January 2018, p.343–348, 2015.

JIANG, H.; LIU, C.; LIU, L.; LOMBARDI, F.; HAN, J. A review, classification, and comparative evaluation of approximate arithmetic circuits. **ACM Journal on Emerging Technologies in Computing Systems**, [S.l.], v.13, n.4, p.1–34, 2017.

JING, X.; CHAU, L.-P. An efficient three-step search algorithm for block motion estimation. **IEEE transactions on multimedia**, [S.l.], v.6, n.3, p.435–438, 2004.

(JVET), J. V. E. T. **JVET Document w18370**. Disponível em: <https://jvet.hhi.fraunhofer.de/>, JVET.

KAHNG, A. B.; KANG, S. Accuracy-configurable adder for approximate arithmetic designs. **Proceedings - Design Automation Conference**, [S.l.], p.820–825, 2012.

KARCZEWICZ, M.; HU, N.; TAQUET, J.; CHEN, C.-Y.; MISRA, K.; ANDERSSON, K.; YIN, P.; LU, T.; FRANÇOIS, E.; CHEN, J. VVC in-loop filters. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, n.10, p.3907–3925, 2021.

KARWOWSKI, D. Precise Probability Estimation of Symbols in VVC CABAC Entropy Encoder. **IEEE Access**, [S.l.], v.9, p.65361–65368, 2021.

KEDEM, Z. M.; MOONEY, V. J.; MUNTIMADUGU, K. K.; PALEM, K. V. **Optimizing energy to minimize errors in approximate ripple carry adders**. [S.l.: s.n.], 2011.

KOREN, I. **Computer Arithmetic Algorithms**. 2nd.ed. [S.l.]: A K Peters/CRC Press, 2003.

KUO, C.-M.; KUAN, Y.-H.; HSIEH, C.-H.; LEE, Y.-H. A novel prediction-based directional asymmetric search algorithm for fast block-matching motion estimation. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.19, n.6, p.893–899, 2009.

LAI, Y.-K.; CHEN, L.-F.; HUANG, S.-Y. Hybrid parallel motion estimation architecture based on fast top-winners search algorithm. **IEEE Transactions on Consumer Electronics**, [S.l.], v.56, n.3, p.1837–1842, 2010.

LEDERER, S. **Covid-19's Impact on streaming video**. 1–1p. Disponível em: [https://go.bitmovin.com/hubfs/Covid-19 OTT streaming\\_Bitmovin\\_Analytics\\_Infographic.pdf](https://go.bitmovin.com/hubfs/Covid-19%20OTT%20streaming_Bitmovin_Analytics_Infographic.pdf). Acesso em: 2020-11-26.

LI, J.; ZHANG, S.; YANG, F. Random Forest Accelerated CU Partition for Inter Prediction in H.266/VVC. In: 2013 18TH , 2022. **Anais...** IEEE Computer Society, 2022. v.2022-July.

LI, R.; ZENG, B.; LIOU, M. L. A new three-step search algorithm for block motion estimation. **IEEE transactions on circuits and systems for video technology**, [S.l.], v.4, n.4, p.438–442, 1994.

LIMA, V. H.; STIGGER, M. F.; SOARES, L. B.; DINIZ, C. M.; BAMPI, S. Configurable Approximate Hardware Accelerator to Compute SATD and SAD Metrics for Low Power All-Intra High Efficiency Video Coding. In: SBC/SBMICRO/IEEE/ACM SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN (SBCCI), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p.1–6.

LIN, C.; LEOU, J.-J. An adaptive fast full search motion estimation algorithm for H.264. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2005., 2005. **Anais...** [S.l.: s.n.], 2005. p.1493–1496 Vol. 2.

LIU, B.; XIE, N.; ZHANG, R.; YANG, H.; WANG, Z.; FAN, D.; WANG, Z.; LIU, W.; CAI, H. A 3.8-W 10-Keyword Noise-Robust Keyword Spotting Processor Using Symmetric Compressed Ternary-Weight Neural Networks. **IEEE Open Journal of the Solid-State Circuits Society**, [S.l.], v.3, p.185–196, 2023.

LIU, W.; LOMBARDI, F.; SHULTE, M. A retrospective and prospective view of approximate computing [point of view. **Proceedings of the IEEE**, [S.l.], v.108, n.3, p.394–399, 2020.

LOUKIL, H.; MAYET, A. M. Hardware implementation and validation of the fast variable block size motion estimation architecture for HEVC Standard. **Multimedia Tools and Applications**, [S.l.], v.82, n.30, p.46331–46349, 2023.

LUMENCI. **Understanding the Basics of Video Compression**. Accessed: June 14, 2024. Disponível em: <<https://www.lumenci.com/blog/understanding-the-basics-of-video-compression>>.

MACEDO, M.; SOARES, L.; SILVEIRA, B.; DINIZ, C. M.; Da Costa, E. A. Exploring the use of parallel prefix adder topologies into approximate adder circuits. **ICECS 2017 - 24th IEEE International Conference on Electronics, Circuits and Systems**, [S.l.], v.2018-Janua, p.298–301, 2018.

MAHDIANI, H. R.; AHMADI, A.; FAKHRAIE, S. M.; LUCAS, C. Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v.57, n.4, p.850–862, 2010.

Manikantta Reddy, K.; VASANTHA, M. H.; Nithin Kumar, Y. B.; DWIVEDI, D. Design of approximate dividers for error tolerant applications. **Midwest Symposium on Circuits and Systems**, [S.l.], v.2018-Augus, n.1, p.496–499, 2019.

Marcelo Porto. Desenvolvimento Algorítmico e Arquitetural para a Estimação de Movimento na Compressão de Vídeo de Alta Definição. , [S.l.], 2012.

MARSH, J. **How many watts does a phone charger use?** Disponível em: <<https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/>>.

MISHRA, V.; HASSAN, N.; MEHTA, A.; CHATTERJEE, U. DARK-Adders: Digital Hardware Trojan Attack on Block-based Approximate Adders. In: INTERNATIONAL CONFERENCE ON VLSI DESIGN AND 2023 22ND INTERNATIONAL CONFERENCE ON EMBEDDED SYSTEMS (VLSID), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.371–376.

MORIN, L. Intra Coding Tools for Versatile Video Coding ( VVC ). , [S.l.], n.March, 2020.

MUTHULAKSHMI, S.; DASH, C. S.; PRABAHARAN, S. R. Memristor augmented approximate adders and subtractors for image processing applications: An approach. **AEU - International Journal of Electronics and Communications**, [S.l.], v.91, p.91–102, 2018.

PAKDAMAN, F.; ADELIMANESH, M. A.; GABBOUJ, M.; HASHEMI, M. R. COMPLEXITY ANALYSIS OF NEXT-GENERATION VVC ENCODING AND DECODING 1 School of Electrical and Computer Engineering , University of Tehran , Tehran , Iran 2 Department of Computing Sciences , Tampere University , Tampere , Finland. , [S.l.], n.Ld, p.2–7, 2019.

PAKDAMAN, F.; ADELIMANESH, M. A.; GABBOUJ, M.; HASHEMI, M. R. Complexity Analysis Of Next-Generation VVC Encoding And Decoding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2020., 2020. **Anais...** IEEE, 2020. p.3134–3138.

PAKDAMAN, F.; HASHEMI, M. R.; GHANBARI, M. A low complexity and computationally scalable fast motion estimation algorithm for HEVC. **Multimedia Tools and Applications**, [S.l.], v.79, p.11639–11666, 5 2020.

PALTRINIERI, A.; PELOSO, R.; MASERA, G.; SHAFIQUE, M.; MARTINA, M. On the Effect of Approximate-Computing in Motion Estimation. **Journal of Low Power Electronics**, [S.l.], v.15, n.1, p.40–50, 2019.

PARK, D.; YOON, Y.-U.; KIM, J.-G.; LEE, J.; KANG, J.-W. Enhancement on the Entropy Coding of Intra Mode in VVC. In: IEEE INTERNATIONAL SYMPOSIUM ON BROADBAND MULTIMEDIA SYSTEMS AND BROADCASTING (BMSB), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1–3.

PFAFF, J.; FILIPPOV, A.; LIU, S.; ZHAO, X.; CHEN, J.; DE-LUXÁN-HERNÁNDEZ, S.; WIEGAND, T.; RUFITSKIY, V.; RAMASUBRAMONIAN, A. K.; AUWERA, G. Van der. Intra prediction and mode coding in VVC. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, n.10, p.3834–3847, 2021.

PFAFF, J.; HELLE, P.; MERKLE, P.; SCHA, M.; STALLENBERGER, P.; HINZ, T.; SCHWARZ, H.; MARPE, D.; WIEGAND, T. Data-Driven Intra-Prediction Modes in the Development of the Versatile Video Coding Standard. **ITU Journal**, [S.l.], v.3, 2020.

PORTO, M. **Avaliação do Algoritmo Diamond Search para Estimação de Movimento com Múltiplos quadros de Referência**. 48f. Trabalho Individual – Instituto de Informática, UFRGS, Porto Alegre.

PORTO, R.; AGOSTINI, L.; ZATT, B.; PORTO, M.; ROMA, N.; SOUSA, L. Energy-efficient motion estimation with approximate arithmetic. **2017 IEEE 19th International Workshop on Multimedia Signal Processing, MMSP 2017**, [S.l.], v.2017-Janua, p.1–6, 2017.

PORTO, R.; AGOSTINI, L.; ZATT, B.; ROMA, N.; PORTO, M. Power-Efficient Approximate SAD Architecture with LOA Imprecise Adders. In: IEEE 10TH LATIN AMERICAN SYMPOSIUM ON CIRCUITS SYSTEMS (LASCAS), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.65–68.

PORTO, R.; ZATT, B.; ROMA, N.; AGOSTINI, L.; PORTO, M. 2PSA: An Optimized and Flexible Power-Precision Scalable Adder. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN (SBCCI), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–6.

PROJECT, T. S. **"Climate crisis: The Unsustainable Use of Online Video": Our new report**. Disponível em: <<https://theshiftproject.org/en/article/unsustainable-use-online-video/>>.

RAHA, A.; JAYAKUMAR, H.; RAGHUNATHAN, V. A power efficient video encoder using reconfigurable approximate arithmetic units. **Proceedings of the IEEE International Conference on VLSI Design**, [S.l.], p.324–329, 2014.

RATING, W. **Estatísticas e tendências da Internet para 2024**. Acessado em: 14 de junho de 2024. Disponível em: <<https://www.websiterating.com>>.

RAVURI, R. P. Diamond Search Optimization-Based Technique for Motion Estimation in Video Compression. **International Journal of e-Collaboration**, [S.l.], v.19, n.3, 2023.

REDA, S.; SHAFIQUE, M. **Approximate Circuits: Methodologies and CAD**. [S.l.]: Springer, 2018.

REDDY, K. M.; VASANTHA, M. H.; KUMAR, Y. B. N.; DWIVEDI, D. Design of Approximate Dividers for Error Tolerant Applications. **2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)**, [S.l.], n.1, p.496–499, 2018.

RICHARDSON, I. E. H. **264 and MPEG-4 video compression**: video coding for next-generation multimedia. [S.l.]: John Wiley & Sons, 2004.

Roger Endrigo Carvalho Porto. Exploração de Computação Aproximada no Projeto de Hardware Dedicado de Baixo Consumo para Codificação de Vídeo em Dispositivos Móveis. , [S.l.], v.21, n.1, p.1–9, 2020.

SALDANHA, M. R. d. F. Exploration of encoding time reduction solutions for intra-frame prediction of VVC encoders. , [S.l.], 2021.

SEILER, F.; TAHERINEJAD, N. An IMPLY-based Semi-Serial Approximate In-Memristor Adder. In: IEEE NORDIC CIRCUITS AND SYSTEMS CONFERENCE (NORCAS), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.1–7.

SHAFIQUE, M.; AHMAD, W.; HAFIZ, R.; HENKEL, J. A low latency generic accuracy configurable adder. **Proceedings - Design Automation Conference**, [S.l.], v.2015-July, 2015.

SHAFIQUE, M.; HAFIZ, R.; REHMAN, S.; EL-HAROUNI, W.; HENKEL, J. Invited: Cross-layer approximate computing: From logic to architectures. **Proceedings - Design Automation Conference**, [S.l.], v.2016-Augus, 2016.

SHI, Y. Q.; SUN, H. **Image and video compression for multimedia engineering**: Fundamentals, algorithms, and standards. [S.l.]: CRC press, 2008.

SILVEIRA, B.; ABREU, B.; PAIM, G.; GRELLER, M.; FERREIRA, R.; DINIZ, C.; COSTA, E.; BAMPI, S. Using adder and subtractor compressors to sum of absolute transformed differences architecture for low-power video encoding. **ICECS 2017 - 24th IEEE International Conference on Electronics, Circuits and Systems**, [S.l.], v.2018-Janua, p.490–493, 2018.

Siqueira, I.; Correa, G.; Grellert, M. Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders. In: IEEE 11TH LATIN AMERICAN SYMPOSIUM ON CIRCUITS SYSTEMS (LASCAS), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–4.

SJÖBERG, R.; STRÖM, J.; LITWIC, Ł.; ANDERSSON, K. **Versatile Video Coding for a 5G world - Ericsson**. 5p. Disponível em: <[https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/versatile-video-coding-explained?gclid=CjwKCAiAxKv\\_BRBdEiwAyd40N5e9hwo8GuUwAm6ufLvLDMeBaXoEaqlzRk7](https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/versatile-video-coding-explained?gclid=CjwKCAiAxKv_BRBdEiwAyd40N5e9hwo8GuUwAm6ufLvLDMeBaXoEaqlzRk7)>. Acesso em: 2020-12-29.

SPRINGERLINK. **Advanced Methods for Video Compression**. Affine motion compensation for complex video movements. Disponível em: <<https://link.springer.com>>.

STOLL, J. **Number of Netflix paid subscribers worldwide from 1st quarter 2013 to 4th quarter 2020**. Disponível em: <<https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/>>. Acesso em: mar 10, 2021.

Sullivan, G. J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.22, n.12, p.1649–1668, Dec 2012.

Tanou, J. L.; Blestel, M. Analysis of Emerging Video Codecs: Coding Tools, Compression Efficiency. **SMPTE Motion Imaging Journal**, [S.l.], v.128, n.10, p.14–24, 2019.

TAYLOR, P. **Forecast number of mobile users worldwide from 2020 to 2025**. Disponível em: <<https://www.energysage.com/electricity/house-watts/how-many-watts-does-a-phone-charger-use/>>.

TEAM, J. V. E. **Joint Video Experts Team (JVET) Homepage**. Acessado em: 19 de março de 2023. Disponível em: <<https://jvet.hhi.fraunhofer.de/>>.

VENKATARAMANI, S.; CHAKRADHAR, S. T.; ROY, K.; RAGHUNATHAN, A. Approximate computing and the quest for computing efficiency. In: ANNUAL DESIGN AUTOMATION CONFERENCE, 52., 2015. **Proceedings...** [S.l.: s.n.], 2015. p.120.

VERMA, A. K.; BRISK, P.; IENNE, P. Variable latency speculative addition: A new paradigm for arithmetic circuit design. In: DESIGN, AUTOMATION AND TEST IN EUROPE, 2008. **Proceedings...** [S.l.: s.n.], 2008. p.1250–1255.

WEN, C.; WU, Y.; YIN, X.; ZHUO, C. Approximate Floating-Point FFT Design with Wide Precision-Range and High Energy Efficiency. In: ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE (ASP-DAC), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.134–139.

XIE, K.; ZHOU, J.; ZHANG, S.; YANG, F. Fast Inter Prediction Mode Decision Method Based on Random Forest for H.266/VVC. In: 2023 28TH , 2022. **Anais...** Institute of Electrical and Electronics Engineers Inc., 2022.

XU, Q.; MYTKOWICZ, T.; KIM, N. S. Approximate Computing: A Survey. **IEEE Design and Test**, [S.l.], v.33, n.1, p.8–22, 2016.

YOON, Y.-U.; KIM, J.-G. Activity-Based Block Partitioning Decision Method for Versatile Video Coding. **Electronics**, [S.l.], v.11, n.7, p.1061, 2022. Accessed: June 14, 2024.

ZHANG, T.; MAO, S. An Overview of Emerging Video Coding Standards. **GetMobile: Mobile Computing and Communications**, [S.l.], v.22, n.4, p.13–20, 2019.

ZHAO, X.; KIM, S.-H.; ZHAO, Y.; EGILMEZ, H. E.; KOO, M.; LIU, S.; LAINEMA, J.; KARCZEWICZ, M. Transform Coding in the VVC Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, n.10, p.3878–3890, 2021.

ZHU, N.; GOH, W. L.; ZHANG, W.; YEO, K. S.; KONG, Z. H. Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, [S.l.], v.18, n.8, p.1225–1229, 2010.



## **Apêndices**

## APÊNDICE A – Publicações

R. Ferreira, L. Agostini, C. M. Diniz, B. Zatt, *Evaluation of Imprecise Subtractors into Test Zone Search for VVC Encoding*, 36th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI), Rio de Janeiro, Brazil, 2023, pp. 1-6, doi: 10.1109/SBCCI60457.2023.10261956.

R. Ferreira, L. Santos, L. Agostini, C. M. Diniz, B. Zatt, *VVC Interpicture Prediction Using SAD with Imprecise Subtractors: A Quantitative Analysis*, 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Glasgow, United Kingdom, 2022, pp. 1-4, doi: 10.1109/ICECS202256217.2022.9971128.

R. Ferreira, M. Leme, M. Corrêa, L. Agostini, C. Diniz, B. Zatt, *Approximate Subtractor Operator for Low-Power Video Coding Hardware Accelerators*, 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 2019, pp. 426-429, doi: 10.1109/ICECS46596.2019.8964783.

## APÊNDICE B – Análise para Blocos de Tamanhos fixo

### B.1 Contagem de SAD executados: Blocos 16x16

A Tabela 39 apresenta os resultados obtidos na análise dos vídeos BQTerrace e Cactus, por meio do uso do subtrator *baseline*. Esta análise evidencia diferenças significativas na demanda por recursos computacionais durante as diversas fases de codificação entre os dois vídeos. Essa análise detalhada das etapas do algoritmo Test Zone Search (TZS) nos vídeos BQTerrace e Cactus utilizando tamanho de blocos 16x16 revelando as características específicas de cada vídeo influenciam o processo de busca.

No vídeo BQTerrace, a etapa de Preditor Mediano ocupa 4,221% do processo, indicando uma utilização moderada. Isso pode ser atribuído à natureza do vídeo, que talvez apresente movimentos ou mudanças de cena menos abruptos, permitindo previsões mais eficazes. Em contraste, o Cactus tem um percentual menor nesta etapa, 2,851%, o que sugere que suas características, como movimentos rápidos ou cenas complexas, podem demandar ajustes mais frequentes e refinados em outras etapas do algoritmo, diminuindo a eficácia das previsões médias.

BQTerrace mostra uma grande dependência da Busca Inicial, com 82,916% do processo dedicado a esta etapa. Isso sugere que a busca inicial é extremamente eficaz ou necessária, possivelmente devido a movimentos previsíveis ou repetitivos que facilitam a localização de diferenças. Por outro lado, Cactus destina 59,612% a essa etapa, um valor significativamente menor, indicando a necessidade de uma abordagem de busca mais diversificada e menos concentrada na inicial, provavelmente devido à presença de elementos mais dinâmicos ou imprevisíveis no vídeo.

Ambos os vídeos utilizam minimamente a busca de Dois Pontos, com 0,379% em BQTerrace e 0,422% em Cactus. Isso indica que outras etapas do algoritmo geralmente atendem às necessidades de busca, com essa técnica sendo empregada ocasionalmente para ajustes finos, possivelmente em situações específicas que as outras etapas não conseguem resolver completamente.

Em BQTerrace, a Varredura compõe 9,520% do processo, atuando como um complemento à busca inicial para cobrir regiões que não foram efetivamente alcançadas. No Cactus, a porcentagem é significativamente maior, 30,278%, refletindo a necessidade de uma varredura extensiva, talvez devido a variações frequentes de movimento

ou mudanças na aparência das cenas que não são totalmente capturadas nas buscas iniciais.

A etapa de Busca Refinada em BQTerrace representa 2,964% do processo, indicando ajustes menores após as etapas anteriores. Cactus, com 6,838%, mostra uma necessidade maior dessa etapa, refletindo a complexidade maior do conteúdo visual e a necessidade de correções e ajustes mais precisos após as buscas iniciais e varreduras.

Essas diferenças entre BQTerrace e Cactus demonstram como o conteúdo e a dinâmica do vídeo afetam diretamente o emprego das diversas etapas do TZS. Enquanto BQTerrace se beneficia de uma busca inicial forte e requer menos varredura e Busca Refinada devido ao seu conteúdo menos dinâmico, Cactus exige uma abordagem mais equilibrada e diversificada, dependendo significativamente das etapas de varredura e Busca Refinada para alcançar a precisão necessária no processo de busca. Essa análise reforça a importância de adaptar o algoritmo TZS às características específicas do conteúdo processado para otimizar a eficiência e eficácia da busca.

Tabela 39 – Contagem de SAD com Subtrator Exato para tamanhos de bloco 16x16 em cada uma das etapas do TZS

| Etapas                  | BQTerrace     |        | Cactus        |        |
|-------------------------|---------------|--------|---------------|--------|
|                         | QTD           | (%)    | QTD           | (%)    |
| <b>Preditor Mediano</b> | 80.926.857    | 4,221  | 78.763.360    | 2,851  |
| <b>Busca Inicial</b>    | 1.589.877.640 | 82,916 | 1.647.014.316 | 59,612 |
| <b>Dois Pontos</b>      | 7.266.040     | 0,379  | 11.645.985    | 0,422  |
| <b>Varredura</b>        | 182.547.287   | 9,520  | 836.561.117   | 30,278 |
| <b>Refinada</b>         | 56.840.214    | 2,964  | 188.915.525   | 6,838  |
| <b>Total</b>            | 1.917.458.038 | 100    | 2.762.900.303 | 100    |

A análise da eficiência operacional em algoritmos, especialmente no contexto do TZS (eficiência segue o mesmo significado descrito na sessão 5.2.1). Esta importância é evidenciada ao longo da discussão dos resultados obtidos nas diferentes etapas do algoritmo, onde subtratores aproximados foram avaliados em termos de sua capacidade de reduzir o número de operações SAD (Sum of Absolute Differences). As referências a tabelas específicas, como a Tabela 40 para o vídeo BQterrace e a Tabela 41 para o vídeo Cactus, fornecem uma base quantitativa para a análise comparativa dos subtratores.

No estudo do vídeo BQterrace, observou-se uma variação significativa na performance dos subtratores. Destacou-se, por exemplo, o subtrator  $AppS_3$  no cenário do Preditor Mediano, apresentando uma redução de 1,28% nas operações SAD, o que sinaliza uma maior eficiência. Em contraste, o subtrator  $AppS_4$  exibiu um aumento de 0,63% nessas operações, marcando uma eficiência relativamente inferior. Essa dinâmica de desempenho foi também observada em outras etapas, como na Busca Inicial e na etapa de Dois Pontos, com o subtrator  $AXCS1_2$  reduzindo significativamente as

operações, ao passo que o  $AXCS2_4$  as aumentou drasticamente, em particular, um acréscimo de 44,34% na etapa de Dois Pontos.

No vídeo Cactus, a análise revelou um cenário semelhante de variabilidade. O subtrator  $AppS_1$ , por exemplo, não apresentou variação nas fases de Preditor Mediano e Busca Inicial, indicando uma performance equiparável ao subtrator *baseline*. Por outro lado, subtratores como o  $AXCS3_1$  e o  $AXCS2_4$  demonstraram uma diminuição na eficiência, com aumentos nas operações SAD em suas respectivas fases. Em contrapartida, o  $AppS_4$  se sobressaiu, especialmente nas etapas de Dois Pontos e Busca Refinada, sugerindo uma elevada eficiência ao reduzir o número de operações SAD.

Avaliando o desempenho geral, notou-se que subtratores como o  $AXCS1_2$  e o  $AppS_4$  se destacaram pela sua capacidade de otimizar a eficiência computacional, enquanto o  $AXCS2_4$  mostrou-se particularmente ineficiente, evidenciando um aumento geral no número de operações SAD. Esta análise sublinha a relevância de uma escolha metódica de subtratores aproximados, visando maximizar a eficiência operacional e, por extensão, a performance do algoritmo TZS.

Em suma, a análise conduzida realça a variabilidade significativa na eficácia dos subtratores ao longo das diversas etapas do algoritmo TZS. A capacidade de alguns subtratores em reduzir efetivamente o número de operações SAD enquanto outros aumentam esse número ressalta a complexidade na seleção de subtratores. Assim, a escolha adequada destes elementos é fundamental para assegurar a otimização da eficiência computacional, destacando-se o  $AXCS1_2$  e o  $AppS_4$  como exemplares de alta eficiência neste contexto.

Tabela 40 – Comparação da Quantidade de SADs Executadas para blocos 16x16: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066.

| Subtrator | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|-----------|---------------------|------------------|----------------|--------------|-------------|----------|
| $AppS_1$  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| $AppS_2$  | -0,780              | -0,700           | -1,690         | 0,520        | -0,280      | -0,611   |
| $AppS_3$  | -1,280              | -1,010           | -2,610         | 2,970        | 2,080       | -0,610   |
| $AppS_4$  | 0,630               | 1,370            | -1,620         | 17,900       | 12,140      | 3,247    |
| $AXCS1_1$ | -0,360              | -0,420           | -0,970         | 0,090        | -0,530      | -0,389   |
| $AXCS1_2$ | -0,970              | -1,050           | -1,810         | -1,390       | -1,300      | -1,130   |
| $AXCS1_3$ | -0,650              | -0,610           | -1,440         | 0,360        | -0,800      | -0,555   |
| $AXCS1_4$ | 0,170               | 0,490            | -0,640         | 9,230        | 5,590       | 1,463    |
| $AXCS2_1$ | -0,620              | -0,720           | -1,530         | -1,090       | -1,780      | -0,812   |
| $AXCS2_2$ | 0,440               | 0,560            | -0,700         | 3,570        | 1,200       | 0,874    |
| $AXCS2_3$ | 0,330               | 0,680            | -1,070         | 8,110        | 3,910       | 1,476    |
| $AXCS2_4$ | -0,220              | 0,940            | -1,790         | 44,340       | 23,290      | 5,666    |
| $AXCS3_1$ | -0,130              | -0,250           | -1,630         | 0,720        | -1,110      | -0,189   |
| $AXCS3_2$ | -0,370              | -0,370           | -1,180         | 1,340        | -0,700      | -0,236   |
| $AXCS3_3$ | 0,220               | 0,300            | -1,580         | 2,410        | 0,170       | 0,496    |
| $AXCS3_4$ | -0,330              | 0,220            | -2,100         | 10,700       | 7,630       | 1,391    |

Tabela 41 – Comparação da Quantidade de SADs Executadas para blocos 16x16: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | 0,480               | 0,550            | 0,450          | -0,390       | -1,240      | 0,154    |
| <i>AppS</i> <sub>3</sub>  | 0,880               | 1,300            | 0,400          | -0,270       | -1,510      | 0,642    |
| <i>AppS</i> <sub>4</sub>  | 0,450               | 1,360            | -0,130         | -1,340       | -2,290      | 0,274    |
| <i>AXCS</i> <sub>11</sub> | 0,460               | 0,430            | 0,340          | 0,150        | -0,270      | 0,311    |
| <i>AXCS</i> <sub>12</sub> | 0,380               | 0,350            | 0,270          | 0,080        | -0,270      | 0,237    |
| <i>AXCS</i> <sub>13</sub> | 0,570               | 0,730            | 0,820          | 0,230        | -0,400      | 0,514    |
| <i>AXCS</i> <sub>14</sub> | 0,580               | 1,130            | 1,360          | 2,190        | 0,700       | 1,423    |
| <i>AXCS</i> <sub>21</sub> | 0,450               | 0,390            | 0,080          | -0,140       | -0,200      | 0,202    |
| <i>AXCS</i> <sub>22</sub> | 0,100               | 0,330            | 0,300          | -0,100       | -1,190      | 0,092    |
| <i>AXCS</i> <sub>23</sub> | 0,500               | 1,030            | 0,800          | -0,090       | -1,270      | 0,532    |
| <i>AXCS</i> <sub>24</sub> | 0,690               | 2,190            | 0,990          | 7,320        | 3,290       | 3,790    |
| <i>AXCS</i> <sub>31</sub> | 0,950               | 0,860            | 0,500          | 0,120        | -0,900      | 0,544    |
| <i>AXCS</i> <sub>32</sub> | 0,450               | 0,470            | 0,270          | 0,070        | -1,220      | 0,245    |
| <i>AXCS</i> <sub>33</sub> | 0,800               | 1,130            | 0,920          | 0,120        | -1,650      | 0,647    |
| <i>AXCS</i> <sub>34</sub> | 0,900               | 1,650            | 0,350          | -0,270       | -1,540      | 0,850    |

### B.1.1 Análise do Resultado do Resultado do Cálculo do SAD para blocos 16x16

A análise estatística, Tabela 42 apresenta um aumento no desvio padrão, alcançando aproximadamente 484,90. Este crescimento, em comparação com análises anteriores, indica uma maior variabilidade nos resultados, evidenciando diferenças ainda mais acentuadas no desempenho entre os subtratores. Esse fenômeno sugere uma distribuição mais ampla dos valores de desempenho, com algumas performances destacando-se positiva ou negativamente em relação ao conjunto geral.

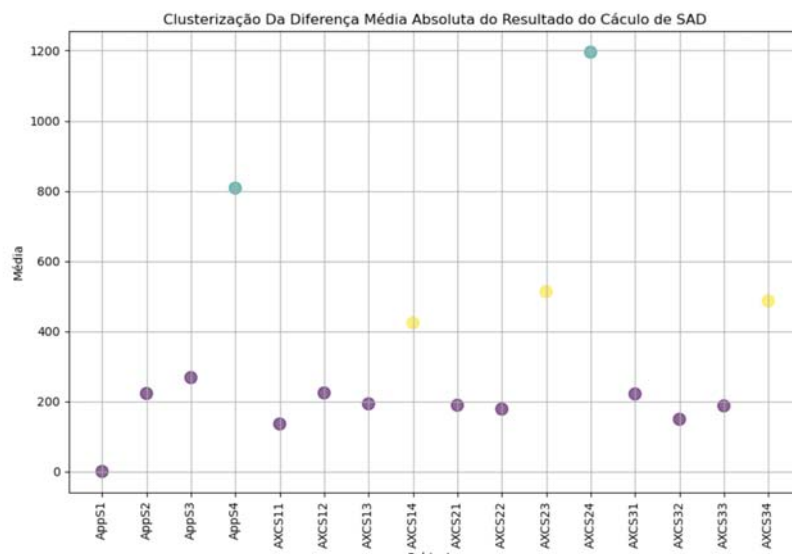
Quanto aos extremos de desempenho, o subtrator *AppS*<sub>1</sub> mantém sua posição como o de melhor desempenho, com um resultado de 0,0. Por outro lado, o *AXCS*<sub>24</sub> identificado como o pior performance, com um resultado de 1993,76.

Tabela 42 – Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 16x16

| Subtrator                 | Média     | Subtrator                 | Média   |
|---------------------------|-----------|---------------------------|---------|
| <i>AppS</i> <sub>1</sub>  | 0,000     | <i>AXCS</i> <sub>11</sub> | 135,380 |
| <i>AppS</i> <sub>2</sub>  | 222,280   | <i>AXCS</i> <sub>12</sub> | 223,870 |
| <i>AppS</i> <sub>3</sub>  | 267,650   | <i>AXCS</i> <sub>13</sub> | 192,880 |
| <i>AppS</i> <sub>4</sub>  | 808,010   | <i>AXCS</i> <sub>14</sub> | 423,650 |
| Subtrator                 | Média     | Subtrator                 | Média   |
| <i>AXCS</i> <sub>21</sub> | 188,830   | <i>AXCS</i> <sub>31</sub> | 220,990 |
| <i>AXCS</i> <sub>22</sub> | 177,950   | <i>AXCS</i> <sub>32</sub> | 148,770 |
| <i>AXCS</i> <sub>23</sub> | 512,860   | <i>AXCS</i> <sub>33</sub> | 187,110 |
| <i>AXCS</i> <sub>24</sub> | 1.195,630 | <i>AXCS</i> <sub>34</sub> | 486,210 |

A análise dos dados, apresentada na Figura 47, está estruturada em três grupos distintos, cada um simbolizado por uma cor específica e por desempenhos variados dos subtratores aproximados em comparação ao subtrator exato. O primeiro grupo, colorido de roxo, compreende a maior parte dos dados e exibe as menores médias de diferença absoluta. Esta característica indica que os subtratores neste *cluster* apresentam um desempenho mais alinhado ao subtrator exato, em contraste com os outros grupos.

Figura 47 – Classificação dos Subtratores por Resultado do Cálculo de SAD. Blocos 16x16



O segundo grupo, ilustrado em amarelo, tem um número menor de pontos que o primeiro e registra médias de diferença absoluta que situam-se entre as dos outros dois grupos. Essa observação sugere que os subtratores aqui classificados têm um desempenho que não alcança o do primeiro grupo, mas que supera o do terceiro.

Por fim, o terceiro grupo, representado em azul claro, engloba menos pontos, mas estes possuem as maiores médias de diferença absoluta, o que indica uma maior discrepância em relação ao desempenho do subtrator exato. Além disso, dentro de cada *cluster* há uma variação significativa nos valores médios de diferença absoluta, como pode ser observado pela distribuição vertical dos pontos.

Identificou-se também um possível outlier no terceiro grupo, cujo valor destaca-se significativamente dos demais. Esse ponto sugere a necessidade de uma análise mais aprofundada para compreender as causas dessa discrepância marcante.

### B.1.2 Análise da Distância do Bloco 16x16

A análise da Tabela 43 sobre o desempenho dos subtratores apresentou informações estatísticas importantes para a compreensão de seu rendimento. A média, calculada em aproximadamente 0,30, serve como um indicador do desempenho médio desses subtratores. Este valor central sugere que, em média, o rendimento observado situou-se neste patamar. A variabilidade desses resultados é quantificada pelo desvio padrão, aproximadamente 0,37. Este valor indica uma dispersão considerável dos resultados em torno da média, apresentando uma heterogeneidade significativa no desempenho dos subtratores. A amplitude dessa variabilidade é indispensável para entender a consistência e a confiabilidade dos subtratores analisados. No espectro de desempenho, o subtrator  $AppS_1$  apresenta resultados iguais ao *baseline*. Em con-

traste, o  $AXCS2_4$  se posiciona como o menos preciso.

Figura 48 – Classificação dos Vetores de Movimento por Subtrator Aproximado. Blocos 16x16

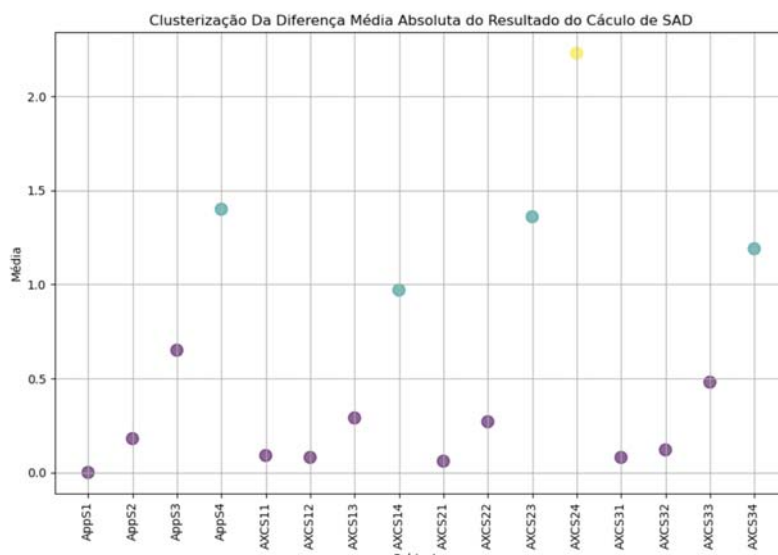


Tabela 43 – Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 16x16

| Subtrator | Média | Subtrator | Média |
|-----------|-------|-----------|-------|
| $AppS_1$  | 0,000 | $AXCS1_1$ | 0,090 |
| $AppS_2$  | 0,180 | $AXCS1_2$ | 0,080 |
| $AppS_3$  | 0,650 | $AXCS1_3$ | 0,290 |
| $AppS_4$  | 1,400 | $AXCS1_4$ | 0,970 |
| Subtrator | Média | Subtrator | Média |
| $AXCS2_1$ | 0,060 | $AXCS3_1$ | 0,080 |
| $AXCS2_2$ | 0,270 | $AXCS3_2$ | 0,120 |
| $AXCS2_3$ | 1,360 | $AXCS3_3$ | 0,480 |
| $AXCS2_4$ | 2,230 | $AXCS3_4$ | 1,190 |

Os dados expostos na Figura 48 são divididos em três clusters, cada um representado por uma cor distinta e por diferentes níveis de desempenho dos subtratores aproximados em relação ao subtrator exato. O primeiro *cluster* em tonalidade roxa, contém a maior quantidade de dados e mostra as menores médias de diferença absoluta, indicando que os subtratores neste grupo estão mais próximos ao desempenho do subtrator exato em comparação aos demais.

O segundo *cluster* representado na cor verde, possui menos pontos do que o primeiro e suas médias de diferença absoluta estão posicionadas entre as dos outros dois grupos. Tal comportamento indica que os subtratores desse grupo desempenham melhor que os do terceiro, porém não tão bem quanto os do primeiro.

O terceiro *cluster* em amarelo claro, apresenta o menor número de pontos, porém com as médias de diferença absoluta mais altas, revelando uma discrepância acentuada em relação ao desempenho do subtrator exato. Observa-se ainda uma variação



considerável nos valores médios de diferença absoluta dentro de cada grupo, como evidenciado pela distribuição vertical dos pontos.

Um possível outlier foi identificado no terceiro *cluster* notadamente distante dos demais pontos, o que aponta para a necessidade de investigações mais detalhadas para entender as razões por trás desta diferença expressiva.

### B.1.3 Eficiência de Codificação para blocos 16x16

A análise dos resultados da eficiência de codificação, conforme ilustrado na Tabela 44, apresenta variações importantes na performance entre diferentes subtratores e operadores para as sequências BQTerrace e Cactus. Observa-se que, para a sequência BQTerrace, o subtrator  $AppS_4$  é menos eficiente, enquanto o  $AXCS_3$  demonstra um aumento na eficiência. Para a sequência Cactus, os subtratores  $AXCS_1$  e  $AXCS_4$  são associados a melhorias na eficiência, contrastando com o  $AXCS_2$ , que apresenta uma eficiência reduzida.

Os resultados indicam que a eficácia dos subtratores  $AppS$  varia; na sequência BQTerrace, o  $AppS_1$  exibe um desempenho neutro, e o  $AppS_4$ , o mais baixo desempenho. Em contraste, na sequência Cactus, esses subtratores tendem a apresentar melhores desempenhos, com valores de BD-RATE próximos a zero ou negativos. Os subtratores  $AXCS_1$ ,  $AXCS_2$  e  $AXCS_3$  mostram resultados inconsistentes, não apresentando uma tendência clara de melhoria ou deterioração na eficiência.

A análise realizada sobre o impacto dos subtratores na codificação de vídeo sublinha que, apesar de sua influência significativa na precisão e confiabilidade do cálculo do *Sum of Absolute Differences* (SAD), o efeito na eficiência geral de codificação é relativamente limitado.

Tabela 44 – Eficiência BD-RATE. Blocos 16x16

| Vídeo:BQTerrace |          |          |          |          |
|-----------------|----------|----------|----------|----------|
| BitsAproximado  | AppS     | AXC1     | AXCS2    | AXCs3    |
| 1               | 0,000    | -0,066   | 0,048    | 0,0241   |
| 2               | 0,0582   | 0,0529   | 0,0347   | 0,1091   |
| 3               | -0,0185  | 0,0229   | -0,0034  | -0,121   |
| 4               | 0,115    | -0,0334  | 0,1046   | -0,0153  |
| Vídeo:Cactus    |          |          |          |          |
| BitsAproximado  | AppS     | AXCS1    | AXCS2    | AXCS3    |
| 1               | 0,000    | -0,00094 | 0,07645  | -0,00443 |
| 2               | -0,02382 | -0,03216 | -0,0452  | -0,01511 |
| 3               | -0,02323 | -0,07497 | 0,07544  | 0,00282  |
| 4               | -0,00186 | -0,1206  | -0,03935 | -0,03686 |

## B.2 Contagem de SAD executados: Blocos 64x64

A Tabela 45 apresenta os resultados da contagem de SAD para cada etapa do TZS (*Test Zone Search*), empregando o subtrator exato nos vídeos BQTerrace e Cactus. Essa análise detalha o comportamento da soma das diferenças absolutas (SAD), evidenciando diferenças significativas entre as etapas de processamento dos dois vídeos. Para os vídeos BQTerrace e Cactus revela informações importantes sobre a aplicação do algoritmo em diferentes tipos de conteúdo visual. Primeiramente, ao considerar o Preditor Mediano (PM), observa-se que BQTerrace utiliza essa etapa de forma mais intensa em comparação com Cactus. Isso sugere que BQTerrace possa ter características mais previsíveis, como padrões de movimento ou estruturas visuais, que são melhor capturadas ou previstas por esta etapa. Por outro lado, a menor utilização em Cactus pode indicar diferenças na eficácia do preditor ou na natureza do conteúdo do vídeo.

A Busca Inicial (BI) emerge como uma etapa crucial, com BQTerrace apresentando uma predominância significativa em comparação com Cactus. Isso indica que BQTerrace se beneficia de uma busca inicial forte, possivelmente devido a movimentos mais consistentes ou repetitivos que são mais facilmente identificados numa busca ampla. Por outro lado, a menor proporção de Busca Inicial em Cactus pode refletir uma complexidade maior ou variações mais frequentes no movimento que desafiam a busca inicial.

Quanto à etapa de Dois Pontos (DP), ambos os vídeos mostram uma utilização muito baixa, sugerindo que a busca em dois pontos não é tão eficaz ou necessária para esses conteúdos específicos. No que diz respeito à Varredura, observa-se que Cactus depende significativamente mais dessa etapa em comparação com BQTerrace. Isso pode ser atribuído a uma necessidade de explorar mais detalhadamente o espaço de busca, possivelmente devido à presença de movimentos rápidos ou irregulares que não foram totalmente resolvidos nas etapas anteriores. Em contrapartida, BQTerrace, com uma varredura menor, parece lidar com o conteúdo de forma mais eficiente nas etapas iniciais.

Por fim, a etapa de Busca Refinada apresenta uma diferença marcante entre os vídeos, com Cactus apresentando mais do que o dobro da observada em BQTerrace. Isso sugere que Cactus requer ajustes mais finos após as etapas iniciais e de varredura, possivelmente devido a desafios na captura de detalhes finos ou na adaptação a mudanças rápidas no cenário visual.

Em conclusão, a análise detalhada das etapas do TZS reflete como as características intrínsecas de cada vídeo afetam a aplicação do algoritmo. BQTerrace parece beneficiar-se de uma forte busca inicial, com menos necessidade de Busca Refinada posteriores, enquanto Cactus desafia o algoritmo com a necessidade de intensas var-

reduras e Busca Refinada, indicando um conteúdo potencialmente mais complexo e dinâmico.

A investigação detalhada das Tabelas 46 e 47 oferece uma visão comparativa entre os subtratores aproximados e o subtrator *baseline*, utilizando blocos de 64x64. A análise foca em nuances importantes no desempenho dos subtratores dentro do algoritmo Test Zone Search (TZS), especialmente nas variações da eficiência operacional em etapas distintas do processo, como descrito na sessão 5.2.1. A avaliação quantifica a habilidade de cada subtrator em modificar o número de operações Soma das Diferenças Absolutas (SAD) em relação a um subtrator exato.

Ao analisar as tabelas que detalham o desempenho do algoritmo Test Zone Search (TZS) nos vídeos BQTerrace e Cactus, identificamos uma diferença significativa nas demandas de processamento de cada um. O BQTerrace se destaca pelo alto número de operações SAD, particularmente nas etapas de Varredura e Busca Refinada. Esta tendência indica uma maior complexidade ou desafios mais acentuados que exigem ajustes frequentes e detalhados para manter a precisão.

Em contraste, o Cactus, embora registre um maior número de operações na etapa de Dois Pontos (DP), apresenta menor atividade nas fases subsequentes, sugerindo uma complexidade geral reduzida ou maior eficácia dos subtratores nas etapas finais do algoritmo. Este padrão implica que menos correções são necessárias à medida que o processo avança, otimizando o uso dos recursos computacionais.

Além disso, a análise do total de operações SAD reforça a noção de que o BQTerrace é mais exigente em termos de processamento. Esse vídeo requer quase quatro vezes mais operações em comparação ao Cactus, sublinhando as diferenças em termos de desafios e complexidade enfrentados pelo TZS em cada caso.

Aprofundando na performance dos diferentes subtratores em cada etapa, percebemos variações significativas. Subtratores exatos, como o  $AppS_1$ , não mostram diferenças nas operações SAD, servindo de referência. Por outro lado, subtratores graduais como  $AppS_2$ ,  $AppS_3$  e  $AppS_4$ , destacam-se especialmente nas etapas mais críticas de ajuste, evidenciando a necessidade de adaptações na configuração para otimizar a resposta do algoritmo às características específicas de cada vídeo.

As séries AXCS também mostram comportamentos distintos. Enquanto a série AXCS1 e AXCS2 variam de mudanças moderadas a extremas nas operações SAD, o  $AXCS3_1$  da série AXCS3 é notável por suas eficiências em reduzir a necessidade de operações, sugerindo um alinhamento superior com as exigências de processamento do vídeo em que foi testado.

Essas descobertas são cruciais para entender como pequenas alterações na configuração dos subtratores podem influenciar significativamente a eficiência do TZS, reforçando a importância da escolha adequada do subtrator, não apenas baseada no desempenho geral, mas também na sua adequação às demandas específicas de cada

etapa do processo e do vídeo em questão.

Tabela 45 – Contagem de SAD com Subtrator Exato para blocos 64x64 em cada uma das etapas do TZS

| Etapas                  | BQTerrace  |         | Cactus     |         |
|-------------------------|------------|---------|------------|---------|
|                         | QTD        | (%)     | QTD        | (%)     |
| <b>Preditor Mediano</b> | 1.931.819  | 4,241   | 2.451.631  | 2,765   |
| <b>Busca Inicial</b>    | 37.812.855 | 83,004  | 50.226.953 | 56,644  |
| <b>Dois Pontos</b>      | 189.633    | 0,416   | 306.855    | 0,346   |
| <b>Varredura</b>        | 4.298.629  | 9,436   | 29.323.188 | 33,070  |
| <b>Refinada</b>         | 1.322.514  | 2,903   | 6.362.598  | 7,175   |
| <b>Total</b>            | 45.555.450 | 100,000 | 88.671.226 | 100,000 |

Tabela 46 – Comparação da Quantidade de SADs Executadas para blocos 64x64: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066

| Subtrator                | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|--------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS<sub>1</sub></i>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS<sub>2</sub></i>  | 0,691               | 0,980            | 1,990          | 6,200        | 5,820       | 1,634    |
| <i>AppS<sub>3</sub></i>  | 0,160               | 0,920            | 2,870          | 24,700       | 17,160      | 3,618    |
| <i>AppS<sub>4</sub></i>  | 0,720               | 2,570            | 4,650          | 101,020      | 64,480      | 13,618   |
| <i>AXCS<sub>11</sub></i> | 0,090               | 0,120            | -0,110         | 1,370        | 1,440       | 0,278    |
| <i>AXCS<sub>12</sub></i> | -0,070              | -0,090           | 0,540          | 1,970        | 0,210       | 0,114    |
| <i>AXCS<sub>13</sub></i> | 0,270               | 0,660            | 1,760          | 13,010       | 7,960       | 2,037    |
| <i>AXCS<sub>14</sub></i> | 0,140               | 1,320            | 5,100          | 56,800       | 31,390      | 7,400    |
| <i>AXCS<sub>21</sub></i> | 0,260               | 0,260            | 0,360          | 0,350        | -0,970      | 0,244    |
| <i>AXCS<sub>22</sub></i> | 0,650               | 1,100            | 2,570          | 13,530       | 10,430      | 2,559    |
| <i>AXCS<sub>23</sub></i> | 0,560               | 1,560            | 1,900          | 63,510       | 31,130      | 8,247    |
| <i>AXCS<sub>24</sub></i> | 0,190               | 2,400            | 1,930          | 214,730      | 91,590      | 24,937   |
| <i>AXCS<sub>31</sub></i> | -0,010              | -0,050           | 0,120          | 1,330        | -1,400      | 0,043    |
| <i>AXCS<sub>32</sub></i> | 0,910               | 1,060            | 2,030          | 2,690        | 1,700       | 1,269    |
| <i>AXCS<sub>33</sub></i> | 0,520               | 1,110            | 2,640          | 18,720       | 13,210      | 3,126    |
| <i>AXCS<sub>34</sub></i> | 0,030               | 1,400            | 2,040          | 74,500       | 45,710      | 9,530    |

### B.2.1 Análise do Resultado do Resultado do Cálculo do SAD para blocos 64x64

Esta análise, detalhada na Tabela 24, foca nos resultados dos subtratores aproximados em comparação com o subtrator exato, analisando o resultado do *Sum of Absolute Differences* (SAD). A média é de 0,424, indicando um desempenho mediano dos subtratores, onde, tipicamente, os valores concentram-se próximo a essa marca. No entanto, o desvio padrão, identificado em 0.503, apresenta uma ampla dispersão dos dados ao redor da média, apontando para uma significativa variabilidade nos resultados dos subtratores.

Dentre os subtratores avaliados, o *AppS<sub>1</sub>* destaca-se por performar igual ao subtrator exato, alcançando 0,00 de média, o que, conforme estabelecido, representa o ideal de desempenho. Em contrapartida, o *AXCS<sub>24</sub>* se destaca com a menor precisão, com uma média de 1,82%, o mais elevado entre os subtratores analisados, denotando assim o pior resultado.

Os dados apresentados na Figura 49 são organizados em quatro clusters distintos,

Tabela 47 – Comparação da Quantidade de SADs Executadas para blocos 32x32: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | 1,060               | 1,300            | 2,620          | 1,160        | 1,340       | 1,284    |
| <i>AppS</i> <sub>3</sub>  | 0,340               | 1,010            | 4,790          | 0,470        | 0,530       | 0,801    |
| <i>AppS</i> <sub>4</sub>  | 0,100               | 1,530            | 10,500         | 2,200        | 2,760       | 1,834    |
| <i>AXCS</i> <sub>11</sub> | 0,170               | 0,120            | 0,010          | -0,710       | -0,230      | -0,174   |
| <i>AXCS</i> <sub>12</sub> | 0,340               | 0,310            | 0,100          | 0,100        | 0,420       | 0,258    |
| <i>AXCS</i> <sub>13</sub> | 0,770               | 1,030            | 2,760          | 0,900        | 0,970       | 1,003    |
| <i>AXCS</i> <sub>14</sub> | 0,380               | 1,440            | 8,230          | 5,200        | 5,000       | 2,944    |
| <i>AXCS</i> <sub>21</sub> | 0,130               | 0,100            | 0,210          | -0,260       | 0,000       | -0,021   |
| <i>AXCS</i> <sub>22</sub> | 0,580               | 0,930            | 3,430          | -0,030       | 0,450       | 0,593    |
| <i>AXCS</i> <sub>23</sub> | -0,160              | 0,760            | 4,950          | 3,760        | 2,620       | 1,870    |
| <i>AXCS</i> <sub>24</sub> | 0,250               | 2,480            | 12,060         | 16,260       | 12,440      | 7,730    |
| <i>AXCS</i> <sub>31</sub> | -0,150              | -0,140           | -0,110         | -0,760       | -0,570      | -0,380   |
| <i>AXCS</i> <sub>32</sub> | 0,310               | 0,430            | 0,910          | -0,240       | -0,090      | 0,178    |
| <i>AXCS</i> <sub>33</sub> | 0,340               | 0,880            | 4,110          | 0,540        | 0,590       | 0,752    |
| <i>AXCS</i> <sub>34</sub> | 0,420               | 1,680            | 9,070          | 2,510        | 3,000       | 2,052    |

Tabela 48 – Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 64x64

| Subtrator                 | Média     | Subtrator                 | Média     |
|---------------------------|-----------|---------------------------|-----------|
| <i>AppS</i> <sub>1</sub>  | 0,000     | <i>AXCS</i> <sub>11</sub> | 1.568,450 |
| <i>AppS</i> <sub>2</sub>  | 1.568,540 | <i>AXCS</i> <sub>12</sub> | 1.634,670 |
| <i>AppS</i> <sub>3</sub>  | 2.517,260 | <i>AXCS</i> <sub>13</sub> | 1.330,760 |
| <i>AppS</i> <sub>4</sub>  | 6.904,750 | <i>AXCS</i> <sub>14</sub> | 2.675,280 |
| Subtrator                 | Média     | Subtrator                 | Média     |
| <i>AXCS</i> <sub>21</sub> | 1.257,740 | <i>AXCS</i> <sub>31</sub> | 1.967,720 |
| <i>AXCS</i> <sub>22</sub> | 1.804,160 | <i>AXCS</i> <sub>32</sub> | 1.510,350 |
| <i>AXCS</i> <sub>23</sub> | 2.921,120 | <i>AXCS</i> <sub>33</sub> | 2.656,130 |
| <i>AXCS</i> <sub>24</sub> | 7.437,020 | <i>AXCS</i> <sub>34</sub> | 3.939,400 |

cada um caracterizado por uma cor específica e por diferentes desempenhos dos subtratores aproximados em relação ao subtrator exato.

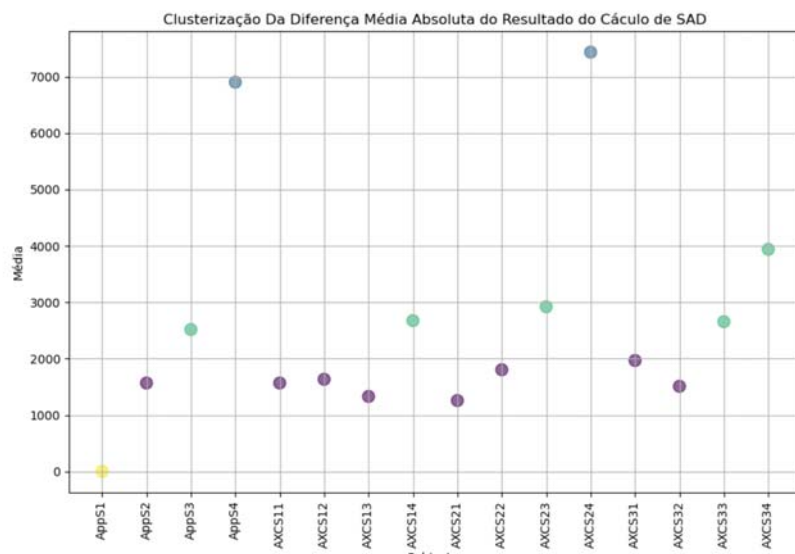
O primeiro *cluster* em tonalidade amarela, é composto por apenas um subtrator. Este subtrator apresenta um desempenho equivalente ao do subtrator *baseline*, estabelecendo um padrão de referência para os demais clusters.

Em seguida, o segundo *cluster* representado na cor roxa, inclui um número maior de pontos em comparação ao primeiro. Suas médias de diferença absoluta são próximas às do primeiro *cluster* sugerindo um desempenho similar. Isso indica que os subtratores deste grupo têm um desempenho superior ao do terceiro *cluster* embora não alcancem a eficácia do subtrator do primeiro cluster.

O terceiro *cluster* colorido de verde claro, possui menos pontos que o segundo, mas as médias de diferença absoluta são as mais elevadas entre os grupos, indicando uma maior discrepância em relação ao desempenho do subtrator exato. Há também uma variação significativa nos valores médios de diferença absoluta dentro deste grupo, conforme ilustrado pela distribuição vertical dos pontos.

Finalmente, o quarto *cluster* em azul, contém os *outliers*. Os subtratores neste grupo apresentam as maiores diferenças em relação aos demais, destacando-se por

Figura 49 – Classificação dos Subtratores por Resultado do Cálculo de SAD. Blocos 64x64



um desempenho significativamente divergente.

### B.2.2 Análise Distância para Blocos 64x64

A análise da Tabela 49 apresenta uma média registrada de 0,42, com um desvio padrão de 0,50, refletindo uma variação significativa entre os resultados dos subtratores. Esses indicadores estatísticos são essenciais para avaliar a eficiência dos subtratores em questão.

Em termos de desempenho específico, o  $AppS_1$  se destaca com resultado 0,00. Em contraste, o  $AXCS2_4$  registra o pior caso, com 1,82, marcando-o como o subtrator de pior desempenho no conjunto analisado. Esta diferença entre o melhor e o pior desempenho evidencia a ampla variabilidade entre os subtratores e sublinha a importância de otimizações direcionadas para melhorar a eficácia dos processos de subtração. Essa análise atualizada confirma as tendências anteriormente observadas. O  $AppS_1$  mantém o melhor resultado, indicando um alinhamento perfeito com o subtrator *baseline* e demonstrando sua eficiência e confiabilidade. Em contrapartida, o  $AXCS2_4$  mostra consistentemente o piores resultado, em relação ao *baseline*.

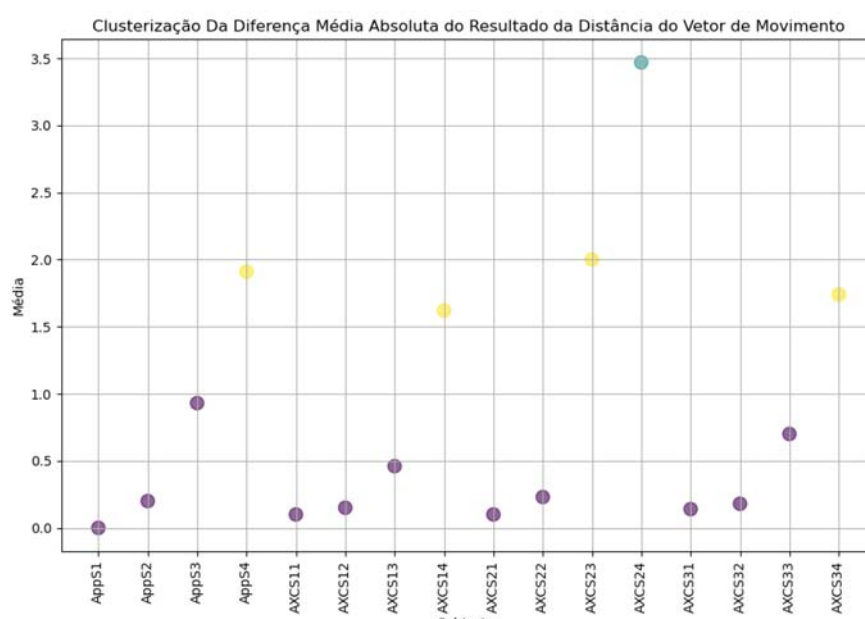
Essa análise atualizada confirma as tendências anteriormente observadas. O  $AppS_1$  mantém o melhor resultado, indicando um alinhamento perfeito com o subtrator *baseline* e demonstrando sua efetividade e confiabilidade. Em contrapartida, o  $AXCS2_4$  mostra consistentemente o pior resultado, em relação ao *baseline*.

A Figura 50 categoriza os dados em três *clusters*, cada um diferenciado por uma cor específica e por distintos níveis de desempenho dos subtratores aproximados em relação ao subtrator exato, permitindo uma análise detalhada do comportamento de cada grupo.

Tabela 49 – Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 64x64

| Subtrator | Média | Subtrator | Média |
|-----------|-------|-----------|-------|
| $AppS_1$  | 0,000 | $AXCS1_1$ | 0,100 |
| $AppS_2$  | 0,200 | $AXCS1_2$ | 0,150 |
| $AppS_3$  | 0,930 | $AXCS1_3$ | 0,460 |
| $AppS_4$  | 1,910 | $AXCS1_4$ | 1,620 |
| Subtrator | Média | Subtrator | Média |
| $AXCS2_1$ | 0,100 | $AXCS3_1$ | 0,140 |
| $AXCS2_2$ | 0,230 | $AXCS3_2$ | 0,180 |
| $AXCS2_3$ | 2,000 | $AXCS3_3$ | 0,700 |
| $AXCS2_4$ | 3,470 | $AXCS3_4$ | 1,740 |

Figura 50 – Classificação dos Vetores de Movimento por Subtrator Aproximado. Blocos 64x64



O primeiro *cluster*, colorido de roxo, engloba a maior parte dos dados e apresenta as menores médias de diferença absoluta entre os grupos. Esta característica indica que os subtratores deste grupo alcançam um desempenho que é mais próximo ao do subtrator exato, sugerindo uma maior precisão desses subtratores em comparação aos demais. Este grupo é composto por uma quantidade significativa de subtratores, destacando-se pela sua proximidade em performance ao padrão exato.

O segundo *cluster*, representado na cor verde, contém menos pontos que o primeiro, situando suas médias de diferença absoluta entre as do primeiro e do terceiro grupo. Esse posicionamento intermediário revela que, embora os subtratores deste grupo não atinjam o nível de precisão do primeiro, eles superam em desempenho os do último *cluster*. Este grupo é formado por apenas quatro subtratores, indicando uma seleção mais restrita mas ainda eficiente.

Já o terceiro *cluster*, em amarelo claro, é o que possui menos pontos, com apenas um subtrator representado. Este subtrator, no entanto, registra a maior média de



diferença absoluta, destacando uma discrepância marcante em relação ao desempenho do subtrator exato. A presença de uma variação considerável nos valores médios de diferença absoluta é observada, principalmente devido à distribuição vertical dos pontos, ressaltando a variabilidade dentro deste grupo.

Ademais, foi identificado um possível *outlier* dentro do terceiro *cluster*, que se destaca notavelmente dos demais pontos. Este fato sugere a necessidade de uma investigação mais aprofundada para compreender as razões dessa discrepância significativa, visando entender melhor os fatores que contribuem para tais variações no desempenho dos subtratores aproximados..

### B.2.3 Eficiência de Codificação para blocos 64x64

Analisando os resultados da Tabela 50 referente ao BD-RATE, podemos destacar alguns pontos interessantes e relevantes. Primeiramente, é importante ressaltar que a Tabela apresenta os resultados do BD-RATE para dois vídeos diferentes, BQTerrace e Cactus, e para cada vídeo são avaliados quatro operadores distintos: *AppS*, *AXCS1*, *AXCS2* e *AXSC3*. Cada operador possui quatro variações, indicadas pelos números 1 a 4.

Observando o vídeo BQTerrace, nota-se que o operador *AppS*<sub>1</sub> apresenta um BD-RATE de 0, ou seja, ele atua como um *baseline* para os outros operadores. Para os demais operadores e variações de *AppS*, percebe-se uma tendência geral de valores negativos de BD-RATE. Isso indica que, em comparação com o *baseline*, essas configurações geraram uma redução na taxa de bits, o que é desejável em muitas aplicações.

O operador *AXCS23* apresenta o menor BD-RATE para o vídeo BQTerrace, com um valor de -0,195, seguido de perto pelos operadores *AXCS34* e *AXCS33*, com valores de -0,184 e -0,177, respectivamente. Esses resultados sugerem que essas configurações podem ser mais eficientes na redução da taxa de bits para o vídeo BQTerrace em comparação com os outros operadores e variações.

Em contraste, no vídeo Cactus, a maioria dos operadores e variações apresenta valores positivos de BD-RATE, com exceção do operador *AXCS34*, que possui um valor de -0,0180. Isso indica que, para o vídeo Cactus, a maioria das configurações testadas não foi tão eficiente na redução da taxa de bits em comparação com o *baseline*. A variação *AXCS24* apresenta o maior BD-RATE, com um valor de 0,0840.

### B.2.4 Contagem de SAD executados: Blocos 128x128

A análise dos valores de SADS da Tabela 51 apresenta diferenças significativas nas etapas do TZS para os vídeos BQTerrace e Cactus.

No vídeo BQTerrace, a etapa de Preditor Mediano é utilizada em 3,552% dos casos, o que é mais intensivo em comparação com 2,802% no vídeo Cactus. Essa



| Btis<br>Proximado | BQTerrace |         |         |         |
|-------------------|-----------|---------|---------|---------|
|                   | AppS      | AXCS1   | AXCS2   | AXCS3   |
| 1                 | 0,0000    | -0,1481 | 0,0228  | -0,0672 |
| 2                 | -0,0515   | -0,0601 | -0,1191 | -0,0363 |
| 3                 | -0,1718   | -0,0779 | -0,1953 | -0,1775 |
| 4                 | -0,1541   | -0,1349 | -0,1462 | -0,1845 |

| Btis<br>Proximado | Cactus |        |        |         |
|-------------------|--------|--------|--------|---------|
|                   | AppS   | AXCS1  | AXCS2  | AXCS3   |
| 1                 | 0,0000 | 0,0075 | 0,0210 | 0,0319  |
| 2                 | 0,0823 | 0,0036 | 0,0789 | 0,0148  |
| 3                 | 0,0774 | 0,0784 | 0,0023 | 0,0481  |
| 4                 | 0,0519 | 0,0553 | 0,0840 | -0,0180 |

Tabela 50 – Eficiência BD-RATE. Blocos 64x64

diferença sugere que BQTerrace, possivelmente devido a movimentos mais previsíveis ou repetitivos, beneficia-se mais dessa fase inicial que se baseia em previsões de movimento ou características anteriores. Por outro lado, a Busca Inicial é extremamente dominante em BQTerrace, ocupando 69,952% do processo, indicativo de uma forte dependência dessa etapa que pode ser devido a movimentos ou variações de intensidade que são consistentes e previsíveis. Em contraste, Cactus apresenta apenas 55,926 nesta etapa, sugerindo que a busca inicial é menos eficaz ou que o vídeo requer abordagens mais diversificadas nas fases subsequentes devido a uma maior dinâmica ou complexidade nos conteúdos apresentados.

Ambos os vídeos mostram um uso mínimo da técnica de Dois Pontos, com BQTerrace utilizando 0,354% e Cactus 0,257%. Isso reflete que a busca em dois pontos específicos pode não ser tão relevante ou eficaz para estes vídeos, possivelmente porque os vetores de movimento não sejam facilmente isoláveis em apenas dois pontos ou que outras técnicas de busca sejam mais adequadas. Em relação à Varredura, existe um contraste marcante entre os dois vídeos: BQTerrace utiliza 21,559% desta etapa, enquanto Cactus necessita de uma varredura mais intensa, com 34,546% dedicados a esta fase. Tal contraste indica que Cactus enfrenta uma complexidade ou variação espacial que requer adaptações no algoritmo para lidar com áreas de movimento irregular ou abrupto, uma característica menos prevalente em BQTerrace.

A Busca Refinada também mostra diferenças significativas, com BQTerrace utilizando 4,583% e Cactus 6,47% desta etapa. O maior uso da Busca Refinada em Cactus reflete a necessidade de ajustes mais detalhados e precisos após as etapas anteriores, indicando imprecisões ou características específicas do vídeo que exigem uma análise mais minuciosa para alcançar a precisão desejada no rastreamento de movimento.

Essa análise detalhada revela como o BQTerrace, com uma forte dependência da Busca Inicial e menor necessidade de Varredura e Busca Refinada, sugere uma previsibilidade ou uniformidade maior. Em contraste, Cactus exige abordagens mais

complexas e diversificadas nas etapas de Varredura e Busca Refinada, indicando uma potencial maior complexidade no conteúdo ou nos movimentos dentro do vídeo.

Tabela 51 – Contagem de SAD com Subtrator Exato para Blocos 128x128 em Cada uma das Etapas do TZS

| Etapas           | BQTerrace |         | Cactus     |         |
|------------------|-----------|---------|------------|---------|
|                  | QTD       | %       | QTD        | %       |
| <b>PM</b>        | 292.207   | 3,552   | 312.292    | 2,802   |
| <b>BI</b>        | 5.753.835 | 69,952  | 6.234.161  | 55,926  |
| <b>DP</b>        | 29.142    | 0,354   | 28.644     | 0,257   |
| <b>Varredura</b> | 1.773.281 | 21,559  | 3.850.890  | 34,546  |
| <b>Ref.</b>      | 376.957   | 4,583   | 721.166    | 6,470   |
| <b>Total</b>     | 8.225.421 | 100,000 | 11.147.152 | 100,000 |

Neste estudo, realizamos uma comparação entre os vídeos BQTerrace e Cactus, ambos codificados com subtratores aproximados, utilizando informações detalhadas nas Tabelas 53 e 52.

No vídeo BQTerrace, a média de -0,17% na etapa do Preditor Mediano sugere uma pequena redução nas operações SAD, indicando uma eficiência ligeiramente melhor nesta fase. Esta é uma das poucas etapas onde o uso de subtratores aproximados mostrou uma vantagem. Com uma média de 0,68% na Busca Inicial, houve um aumento nas operações, refletindo menor eficiência. Isso implica que a etapa de busca inicial não se beneficiou significativamente do uso de subtratores aproximados. A média de 24.56% em Dois Pontos representa o maior aumento entre todas as fases, indicando uma grande perda de eficiência. Essa etapa específica mostrou-se extremamente ineficiente com subtratores aproximados. Um aumento médio de 2.97% nas operações também sugere menor eficiência na etapa de Varredura. Com um aumento médio de 16.36% em Busca Refinada, essa etapa também apresentou uma significativa perda de eficiência, indicando que a Busca Refinada foi bastante afetado pela imprecisão dos subtratores. O aumento médio total de 6,52% nas operações SAD sugere que, de maneira geral, o vídeo BQTerrace teve uma performance pior com o uso de subtratores aproximados, resultando em mais operações do que com o subtrator exato.

No vídeo Cactus, um ligeiro aumento médio de 0,14% nas operações na etapa do Preditor Mediano indica uma eficiência quase neutra nesta etapa. Um aumento moderado de 0,35% também sugere um impacto negativo na Busca Inicial, porém menos pronunciado que em BQTerrace. O aumento de 3,64% em Dois Pontos é substancialmente menor em comparação com o observado em BQTerrace, indicando uma melhor adaptação dos subtratores aproximados nesta etapa para este vídeo. A redução média de -0,68% é notável na etapa de Varredura, pois é a única etapa em que o Cactus mostrou uma melhora na eficiência, reduzindo as operações SAD. O aumento de 1,49% em Busca Refinada é moderado, mostrando que o impacto da

imprecisão foi controlado melhor neste vídeo do que em BQTerrace. Quase nenhuma mudança geral nas operações SAD com uma média de 0,07%, indica que o vídeo "Cactus" manteve uma eficiência comparável ao subtrator exato ao longo do processo.

Essa análise detalhada reforça que a eficácia dos subtratores aproximados varia consideravelmente entre diferentes vídeos e etapas do algoritmo Test Zone Search. Enquanto Cactus apresentou um equilíbrio mais eficiente, BQTerrace sofreu com aumentos significativos nas operações SAD em várias etapas, destacando a importância de considerar o contexto específico de cada aplicação ao avaliar o desempenho dos subtratores aproximados. A análise detalhada por etapas revela tendências importantes sobre como diferentes subtratores impactam as operações do algoritmo Test Zone Search em cada fase específica.

Subtratores como  $AppS_4$  e  $AXCS2_1$  apresentam as maiores reduções nas operações SAD na etapa do Preditor Mediano, indicando eficiência. Em contraste, subtratores como  $AXCS2_4$  mostram aumentos significativos, potencialmente aumentando a carga de processamento nesta fase.  $AXCS2_1$  mostra a maior redução na Busca Inicial, sugerindo uma operação mais eficiente, enquanto  $AXCS2_4$  também lidera os aumentos nesta etapa, o que pode indicar uma escolha menos eficiente para operações que são sensíveis à eficiência na fase de busca inicial.  $AXCS1_1$  destaca-se com uma redução significativa em Dois Pontos, mostrando-se altamente eficiente nesta etapa crítica.  $AXCS1_4$  apresenta o maior aumento, o que sugere um impacto negativo substancial na eficiência do processo nesta etapa.  $AXCS3_3$  e  $AXCS2_1$  lideram as reduções em Varredura, melhorando a eficiência neste componente do algoritmo. Novamente,  $AXCS2_4$  mostra o maior aumento, potencialmente comprometendo a eficiência nesta fase.  $AXCS2_1$  apresenta a maior redução em Busca Refinada, destacando-se como uma opção eficiente para minimizar operações nesta etapa final.  $AXCS1_4$  lidera os aumentos, o que pode ser preocupante para aplicações que exigem alta eficiência no estágio de Busca Refinada.

Os subtratores  $AXCS2_1$  e  $AXCS1_1$  tendem a mostrar desempenhos mais consistentes em termos de redução de operações SAD, sugerindo que são opções eficientes para a maioria das etapas. Em contraste,  $AXCS2_4$  e  $AXCS1_4$  frequentemente mostram os maiores aumentos nas operações SAD, o que pode não ser ideal para aplicações que buscam máxima eficiência.

### **B.2.5 Análise do Resultado do Resultado do Cálculo do SAD para blocos 128x128**

Na análise apresentada, exploramos o desempenho dos subtratores através de um conjunto de dados, utilizando métricas estatísticas para detalhar suas performances. O valor médio, calculado em 5228,85, serve como um ponto de referência para entender o desempenho geral dos subtratores, enquanto o desvio padrão, de 4045,83,

Tabela 52 – Comparação da Quantidade de SADs Executadas para blocos 128x128: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo Cactus, TI: 6,056 e (SI): 1.570,066.

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | 0,070               | 0,010            | -0,020         | -1,120       | -1,960      | -0,504   |
| <i>AppS</i> <sub>3</sub>  | 0,570               | 0,820            | 4,260          | -1,030       | 2,120       | 0,283    |
| <i>AppS</i> <sub>4</sub>  | 0,310               | 0,960            | 14,510         | 0,410        | 3,880       | 0,984    |
| <i>AXCS</i> <sub>11</sub> | -0,110              | -0,120           | 1,060          | -2,660       | -0,240      | -1,005   |
| <i>AXCS</i> <sub>12</sub> | -0,180              | -0,220           | -0,550         | -0,130       | 0,520       | -0,146   |
| <i>AXCS</i> <sub>13</sub> | 0,000               | 0,070            | 3,330          | -3,410       | -0,080      | -1,135   |
| <i>AXCS</i> <sub>14</sub> | 0,120               | 0,730            | 8,250          | 4,070        | 6,690       | 2,275    |
| <i>AXCS</i> <sub>21</sub> | -0,210              | -0,260           | 1,410          | -3,650       | -1,800      | -1,531   |
| <i>AXCS</i> <sub>22</sub> | 0,340               | 0,450            | 2,250          | -3,140       | 0,740       | -0,760   |
| <i>AXCS</i> <sub>23</sub> | 0,220               | 0,550            | 3,680          | 0,200        | 1,770       | 0,513    |
| <i>AXCS</i> <sub>24</sub> | 0,680               | 1,760            | 12,320         | 7,650        | 10,660      | 4,387    |
| <i>AXCS</i> <sub>31</sub> | 0,120               | 0,180            | 1,980          | 0,460        | 0,150       | 0,281    |
| <i>AXCS</i> <sub>32</sub> | 0,040               | 0,110            | 0,450          | -0,150       | 0,410       | 0,040    |
| <i>AXCS</i> <sub>33</sub> | 0,140               | 0,280            | 2,650          | -4,190       | 0,460       | -1,246   |
| <i>AXCS</i> <sub>34</sub> | 0,290               | 0,910            | 10,990         | -0,210       | 4,330       | 0,761    |

Tabela 53 – Comparação da Quantidade de SADs Executadas para blocos 128x128: Subtrator Exato vs. Subtrator Aproximado, Apresentados em Percentuais (%). Valores Negativos Sinalizam uma Redução Benéfica nas Operações. Vídeo BQTerrace, TI: 6,056, SI: 1.570,066.

| Subtrator                 | Preditor Mediano(%) | Busca Inicial(%) | Dois Pontos(%) | Varredura(%) | Refinada(%) | Total(%) |
|---------------------------|---------------------|------------------|----------------|--------------|-------------|----------|
| <i>AppS</i> <sub>1</sub>  | 0,000               | 0,000            | 0,000          | 0,000        | 0,000       | 0,000    |
| <i>AppS</i> <sub>2</sub>  | 0,100               | 0,380            | 0,830          | 2,010        | 6,300       | 0,748    |
| <i>AppS</i> <sub>3</sub>  | -0,370              | 0,300            | 9,880          | 4,060        | 9,590       | 2,767    |
| <i>AppS</i> <sub>4</sub>  | -0,750              | 1,450            | 58,790         | 4,120        | 47,800      | 15,841   |
| <i>AXCS</i> <sub>11</sub> | 0,080               | 0,100            | -5,500         | 2,280        | 0,190       | -1,093   |
| <i>AXCS</i> <sub>12</sub> | 0,050               | 0,070            | -2,620         | 0,670        | 1,710       | -0,432   |
| <i>AXCS</i> <sub>13</sub> | -0,100              | 0,700            | 14,190         | 3,720        | 13,530      | 4,175    |
| <i>AXCS</i> <sub>14</sub> | -0,280              | 1,480            | 67,790         | 8,600        | 37,810      | 17,393   |
| <i>AXCS</i> <sub>21</sub> | -0,120              | -0,130           | -3,110         | 2,110        | 1,710       | -0,684   |
| <i>AXCS</i> <sub>12</sub> | -0,050              | 0,680            | 13,220         | 5,430        | 10,180      | 3,808    |
| <i>AXCS</i> <sub>13</sub> | -0,120              | 1,370            | 52,170         | 1,170        | 25,640      | 13,376   |
| <i>AXCS</i> <sub>14</sub> | 0,040               | 2,930            | 130,250        | 0,990        | 55,640      | 32,686   |
| <i>AXCS</i> <sub>21</sub> | -0,550              | -0,510           | -7,530         | 0,300        | 0,170       | -2,010   |
| <i>AXCS</i> <sub>12</sub> | -0,250              | -0,010           | -5,320         | 3,360        | 4,580       | -0,950   |
| <i>AXCS</i> <sub>13</sub> | 0,010               | 0,760            | 10,430         | 3,670        | 13,200      | 3,399    |
| <i>AXCS</i> <sub>14</sub> | -0,440              | 1,250            | 59,560         | 5,070        | 33,680      | 15,245   |

ilustra a variabilidade dos resultados, indicando uma significativa diferença no desempenho entre eles.

A distribuição dos dados é elucidada pela análise de quartis. O primeiro quartil, situado em 2831,76, mostra que 25% dos subtratores apresentam desempenhos inferiores a esse valor. A mediana, posicionada em 4049,59, divide os dados ao meio, fornecendo uma visão central da distribuição, e o terceiro quartil, em 6918,35, indica que 75% dos subtratores têm desempenhos abaixo desse nível, indicando os 25% mais eficazes acima deste valor. Essa análise destaca a diversidade no rendimento dos subtratores.

O subtrator *AppS*<sub>1</sub>, ao alcançar o resultado perfeito de 0,0, destaca-se por sua performance ideal, não registrando quantidades médias associadas, o que o coloca como o mais eficaz entre todos os analisados. Em contraste, o subtrator *AXCS*<sub>24</sub> apresenta o maior valor registrado, 14448,51, marcando o desempenho mais baixo e

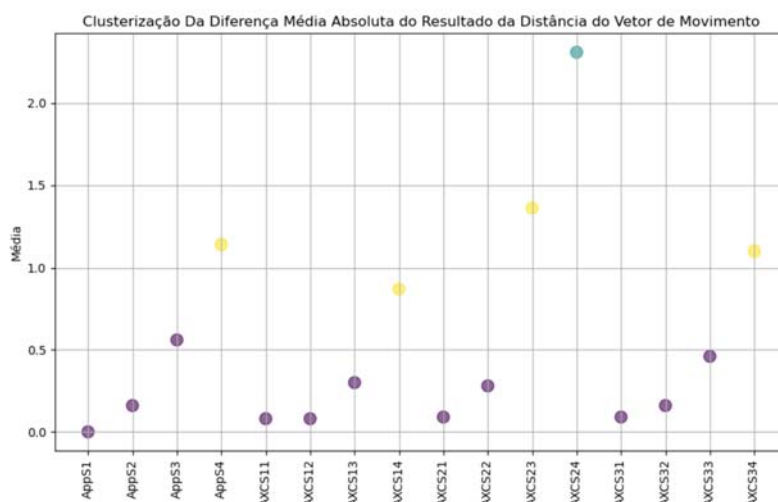
evidenciando as disparidades de performance dentro do grupo estudado.

Através desta avaliação, identificamos não apenas a diversidade de desempenhos entre os subtratores mas também a existência de um subtrator que demonstra alinhamento perfeito com o *baseline*. Esta análise permite uma compreensão mais aprofundada dos níveis de performance.

Tabela 54 – Resultado do Cálculo de SAD: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 128x128

| Subtrator | Média      | Subtrator | Média     |
|-----------|------------|-----------|-----------|
| $AppS_1$  | 0,000      | $AXCS1_1$ | 2.329,550 |
| $AppS_2$  | 2.947,460  | $AXCS1_2$ | 2.484,670 |
| $AppS_3$  | 4.499,950  | $AXCS1_3$ | 3.997,950 |
| $AppS_4$  | 14.008,730 | $AXCS1_4$ | 7.600,330 |
| Subtrator | Média      | Subtrator | Média     |
| $AXCS2_1$ | 3.205,670  | $AXCS3_1$ | 2.100,170 |
| $AXCS2_2$ | 3.011,260  | $AXCS3_2$ | 4.101,230 |
| $AXCS2_3$ | 6.761,540  | $AXCS3_3$ | 4.775,780 |
| $AXCS2_4$ | 14.448,510 | $AXCS3_4$ | 7.388,780 |

Figura 51 – Classificação dos Subtratores por Resultado do Cálculo de SAD. Blocos 128x128



A Figura 51 apresenta os dados em três *clusters*, diferenciados por cor e níveis de desempenho dos subtratores aproximados comparados ao subtrator exato. O primeiro *cluster*, em roxo, contém a maioria dos dados e mostra as menores médias de diferença absoluta, indicando alta precisão e um desempenho próximo ao ideal.

O segundo *cluster*, amarelo, possui menos elementos e desempenho intermediário, com médias de diferença absoluta superiores ao primeiro mas inferiores ao terceiro. Este grupo inclui apenas três subtratores, sugerindo uma seleção eficiente mas limitada.

O terceiro *cluster*, verde claro, é o menor, com apenas um subtrator que exhibe a maior média de diferença absoluta, indicando maior variação de desempenho. A distribuição vertical dos pontos neste grupo ressalta sua volatilidade.

Adicionalmente, um *outlier* observado no terceiro *cluster* merece análise detalhada para entender as causas de sua distinção acentuada e os fatores que afetam o desempenho dos subtratores aproximados.

#### B.2.5.1 Análise de Blocos 128x128 Distância

A análise dos subtratores apresenta um panorama detalhado e distinto quando comparada com avaliações anteriores, refletindo singularidades específicas nas métricas estudadas. Ao focar na coluna média (quantidade média), diversos aspectos do desempenho dos subtratores vêm à tona. A média calculada de 0,565 unidades por subtrator sugere um desempenho padrão moderado, onde, em média, cada subtrator contribui com pouco mais de meio ponto à métrica geral. Esta média é acompanhada de um desvio padrão de 0,639, indicando uma variabilidade moderada nos resultados. Tal dispersão sugere que, embora os valores de média variem, eles não se afastam drasticamente da média, o que denota uma consistência relativamente alta entre os subtratores.

A análise de quartis indica uma perspectiva mais aprofundada sobre a distribuição dos dados. O primeiro quartil, situado em 0,090 unidades, apresenta que um quarto dos subtratores apresenta resultados significativamente baixos, posicionando-se na extremidade inferior do espectro de desempenho. Contrariamente, a mediana, em 0,290 unidades, sinaliza que metade dos subtratores tem desempenhos abaixo dessa marca, o que reflete uma concentração de resultados em torno de valores baixos a moderados. Já o terceiro quartil, estabelecido em 0,9275 unidades, indica que a maior parte dos subtratores não ultrapassa essa métrica, deixando apenas um quarto com desempenhos superiores, os quais representam os resultados mais elevados dentro do conjunto analisado.

No extremo do espectro de desempenho, identificam-se os subtratores que se destacam tanto positiva quanto negativamente. O subtrator  $AppS_1$ , com um valor de 0,0 na média, emerge como o mais eficaz, atingindo o ideal de não contribuir negativamente para a métrica avaliada. Em contrapartida, o subtrator  $AXCS2_4$ , que registra o valor mais alto de 2,31, é marcado como o de pior desempenho no conjunto, destacando-se negativamente em relação aos demais.

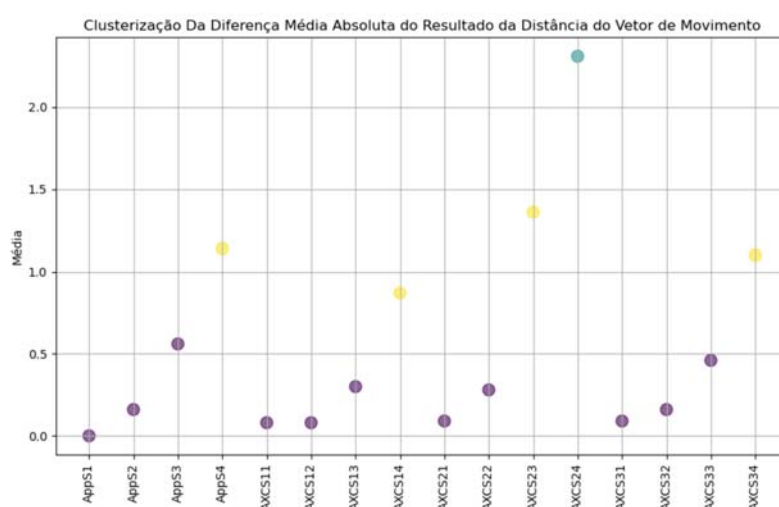
A Figura 52 mencionada ilustra os dados divididos em três grupos, cada um representado por uma cor diferente e por níveis variados de eficácia de certos componentes de cálculo, em comparação com um modelo exato. O primeiro grupo, colorido de roxo, inclui a maior parte dos dados e apresenta as menores diferenças médias em relação ao modelo exato, o que demonstra uma alta precisão e um desempenho muito próximo ao ideal.

O segundo grupo, na cor amarela, tem menos elementos e um desempenho que não é nem tão bom quanto o do primeiro grupo, nem tão ruim quanto o do terceiro, com

Tabela 55 – Distância dos Vetores de Movimento: Diferença Média Absoluta Entre Subtrator Exato e Subtratores Aproximados para blocos 128x128

| Subtrator | Média | Subtrator | Média |
|-----------|-------|-----------|-------|
| $AppS_1$  | 0,000 | $AXCS1_1$ | 0,080 |
| $AppS_2$  | 0,160 | $AXCS1_2$ | 0,080 |
| $AppS_3$  | 0,560 | $AXCS1_3$ | 0,300 |
| $AppS_4$  | 1,140 | $AXCS1_4$ | 0,870 |
| Subtrator | Média | Subtrator | Média |
| $AXCS2_1$ | 0,090 | $AXCS3_1$ | 0,090 |
| $AXCS2_2$ | 0,280 | $AXCS3_2$ | 0,160 |
| $AXCS2_3$ | 1,360 | $AXCS3_3$ | 0,460 |
| $AXCS2_4$ | 2,310 | $AXCS3_4$ | 1,100 |

Figura 52 – Classificação dos Vetores de Movimento por Subtrator Aproximado. Blocos 64x64



diferenças médias maiores que as do primeiro grupo, mas menores que as do terceiro. Esse grupo é composto por apenas três componentes, indicando uma seleção eficaz, porém restrita.

O terceiro grupo, em verde claro, é o menor, contendo apenas um componente que registra a maior média de diferenças, refletindo uma variação maior no desempenho. A disposição vertical dos dados neste grupo destaca a inconsistência do seu desempenho.

Além disso, um ponto atípico observado no terceiro grupo requer uma análise mais aprofundada para compreender as razões da sua distinção marcante e os fatores que influenciam o desempenho dos componentes de cálculo. No entanto, as distâncias entre os grupos são bastante próximas.

### B.2.6 Eficiência de Codificação para blocos 128x128

Ao examinar a Tabela 56 referente aos vídeos BQTerrace e Cactus, observa-se uma análise comparativa do desempenho dos operadores de codificação em distintos tipos de conteúdo de vídeo.

No vídeo BQTerrace, os operadores  $AXCS2_3$  e  $AXCS3_4$  demonstram superiori-

dade em eficiência de codificação, evidenciados por valores negativos de BD-RATE, -0,1953 e -0,184 respectivamente, em comparação ao operador de referência  $AppS_1$ . Isso indica uma melhoria na eficiência. Por outro lado, no vídeo Cactus,  $AXCS1_3$  e  $AXCS2_4$  sobressaem, ainda que seus valores de BD-RATE sejam positivos, 0,078 e 0,084, sugerindo uma eficiência de codificação reduzida frente ao mesmo operador de referência.

Este contraste no desempenho entre diferentes conteúdos de vídeo realça a importância da seleção criteriosa do operador, baseada nas peculiaridades do conteúdo a ser codificado. Além disso, a análise detalhada destes operadores, ressalta a variabilidade e a especificidade de desempenho, evidenciando a necessidade de escolhas informadas para otimizar a eficiência de codificação.

Concluindo, a avaliação dos operadores nos vídeos BQTerrace e Cactus destaca a importância de considerar as características específicas do conteúdo de vídeo na escolha do operador mais adequado.

Tabela 56 – Eficiência BD-RATE. Blocos 128x128

| Btis      |        | BQTerrace |         |         |
|-----------|--------|-----------|---------|---------|
| Proximado | AppS   | AXCS1     | AXCS2   | AXCS3   |
| 1         | 0,0000 | -0,1481   | 0,0228  | -0,0672 |
| 2         |        | -0,0515   | -0,0601 | -0,1191 |
| 3         |        | -0,1718   | -0,0779 | -0,1953 |
| 4         |        | -0,1541   | -0,1349 | -0,1462 |

| Btis      |        | Cactus |        |         |
|-----------|--------|--------|--------|---------|
| Proximado | AppS   | AXCS1  | AXCS2  | AXCS3   |
| 1         | 0,0000 | 0,0075 | 0,0210 | 0,0319  |
| 2         | 0,0823 | 0,0036 | 0,0789 | 0,0148  |
| 3         | 0,0774 | 0,0784 | 0,0023 | 0,0481  |
| 4         | 0,0519 | 0,0553 | 0,0840 | -0,0180 |



### B.3 Máscara de Imprecisão

| Máscara   | Corretas | %     | Bits Imprecisos | Máscara  | Corretas | %     | Bits Imprecisos |
|-----------|----------|-------|-----------------|----------|----------|-------|-----------------|
| PPPPPPPI  | 65.536   | 100   | 1               | PPIPIPIP | 19.200   | 29,3  | 3               |
| PPPPPPPP  | 65.536   | 100   | 0               | IIPPPPPI | 19.008   | 29,00 | 4               |
| PPPPPPPII | 49.152   | 75    | 2               | IIPPPPII | 19.008   | 29,00 | 4               |
| PPPPPPPIP | 49.152   | 75,00 | 1               | IPPPPIII | 19.008   | 29,00 | 4               |
| PPPPPIPI  | 40.960   | 62,50 | 2               | IIPPPPPP | 19.008   | 29,00 | 3               |
| PPPPPIIPP | 40.960   | 62,50 | 1               | IIPPPPIP | 19.008   | 29,00 | 3               |
| PPPPPIII  | 36.864   | 56,25 | 3               | IPPPPIIP | 19.008   | 29,00 | 3               |
| PPPIIPII  | 36.864   | 56,25 | 2               | PIIIPII  | 17.280   | 26,37 | 5               |
| PPPPPIIP  | 36.864   | 56,25 | 2               | PPIIIPI  | 17.280   | 26,37 | 5               |
| PPPIIPPP  | 36.864   | 56,25 | 1               | PPIIPII  | 17.280   | 26,37 | 5               |
| PPPIPPPI  | 34.816   | 53,13 | 2               | PPIPIII  | 17.280   | 26,37 | 5               |
| PPPIPPPP  | 34.816   | 53,13 | 1               | PPIPIII  | 17.280   | 26,37 | 5               |
| PPIPPPPPI | 33.792   | 51,56 | 2               | PIIPIPI  | 17.280   | 26,37 | 4               |
| PPIPPPPP  | 33.792   | 51,56 | 1               | PIIPIPI  | 17.280   | 26,37 | 4               |
| PIPPPPPI  | 33.280   | 50,78 | 2               | PIPIIPI  | 17.280   | 26,37 | 4               |
| PIPPPPPP  | 33.280   | 50,78 | 1               | PIPIPII  | 17.280   | 26,37 | 4               |
| IPPPPPPI  | 33.024   | 50,39 | 2               | PIPIPII  | 17.280   | 26,37 | 4               |
| IPPPPPPP  | 33.024   | 50,39 | 1               | PIPIPII  | 17.280   | 26,37 | 4               |
| PPPIIPI   | 30.720   | 46,88 | 3               | PPIIIIPP | 17.280   | 26,37 | 4               |
| PPPIIPII  | 30.720   | 46,88 | 3               | PPIIIPIP | 17.280   | 26,37 | 4               |
| PPPIIIPP  | 30.720   | 46,88 | 2               | PPIIPIP  | 17.280   | 26,37 | 4               |
| PPPIIPIP  | 30.720   | 46,88 | 2               | PIPIIIP  | 17.280   | 26,37 | 4               |
| PPPIIII   | 27.648   | 42,19 | 4               | PIIPIPP  | 17.280   | 26,37 | 3               |
| PPPIIPI   | 27.648   | 42,19 | 3               | PIIPIPP  | 17.280   | 26,37 | 3               |
| PPPIPII   | 27.648   | 42,19 | 3               | PIIPIPP  | 17.280   | 26,37 | 3               |
| PPPIIIP   | 27.648   | 42,19 | 3               | PIPIPIP  | 17.280   | 26,37 | 3               |
| PPPIIIPP  | 27.648   | 42,19 | 2               | PIPIIIP  | 17.280   | 26,37 | 3               |
| PPPIPIP   | 27.648   | 42,19 | 2               | PIPIPIP  | 17.280   | 26,37 | 3               |
| PPIIIPPI  | 26.112   | 39,84 | 3               | IIPPPPI  | 16.320   | 24,90 | 4               |
| PPIIPPII  | 26.112   | 39,84 | 3               | IIPPPPI  | 16.320   | 24,90 | 4               |
| PPIIIPPP  | 26.112   | 39,84 | 2               | IPIPPPII | 16.320   | 24,90 | 4               |
| PPIPPPI   | 26.112   | 39,84 | 2               | IPIPPPII | 16.320   | 24,90 | 4               |
| PPPIPIPI  | 25.600   | 39,06 | 3               | IPPPPII  | 16.320   | 24,90 | 4               |
| PPPIPIP   | 25.600   | 39,06 | 2               | IPPPPII  | 16.320   | 24,90 | 4               |
| PIIPPPPPI | 25.344   | 38,67 | 3               | IIPPPPP  | 16.320   | 24,90 | 3               |
| PIPPPPPII | 25.344   | 38,67 | 3               | IIPPPIP  | 16.320   | 24,90 | 3               |
| PIIPPPPP  | 25.344   | 38,67 | 2               | IIPPPPP  | 16.320   | 24,90 | 3               |
| PIPPPPPI  | 25.344   | 38,67 | 2               | IPIPPPI  | 16.320   | 24,90 | 3               |
| IIPPPPPPI | 24.960   | 38,09 | 3               | IPPPPIIP | 16.320   | 24,90 | 3               |
| IPPPPPPII | 24.960   | 38,09 | 3               | IPPPPIIP | 16.320   | 24,90 | 3               |
| IIPPPPPP  | 24.960   | 38,09 | 2               | PIPIPII  | 16.000   | 24,41 | 4               |
| IPPPPPPI  | 24.960   | 38,09 | 2               | PIPIPIP  | 16.000   | 24,41 | 3               |
| PPPIIPI   | 23.040   | 35,16 | 4               | PPIIIII  | 15.552   | 23,73 | 6               |
| PPPIIPII  | 23.040   | 35,16 | 4               | PIIIPII  | 15.552   | 23,73 | 5               |
| PPPIPII   | 23.040   | 35,16 | 4               | PIIPIII  | 15.552   | 23,73 | 5               |
| PIPIPIPI  | 23.040   | 35,16 | 3               | PIPIII   | 15.552   | 23,73 | 5               |
| PPIPIPI   | 23.040   | 35,16 | 3               | PPIIIIP  | 15.552   | 23,73 | 5               |
| PPPIIIP   | 23.040   | 35,16 | 3               | IIPPIPI  | 15.552   | 23,73 | 4               |

Tabela 57 – Variação da Imprecisão no Subtrator AppS variando os Bits Imprecisos





## **B.4 Distribuição Entre Etapas TZS**

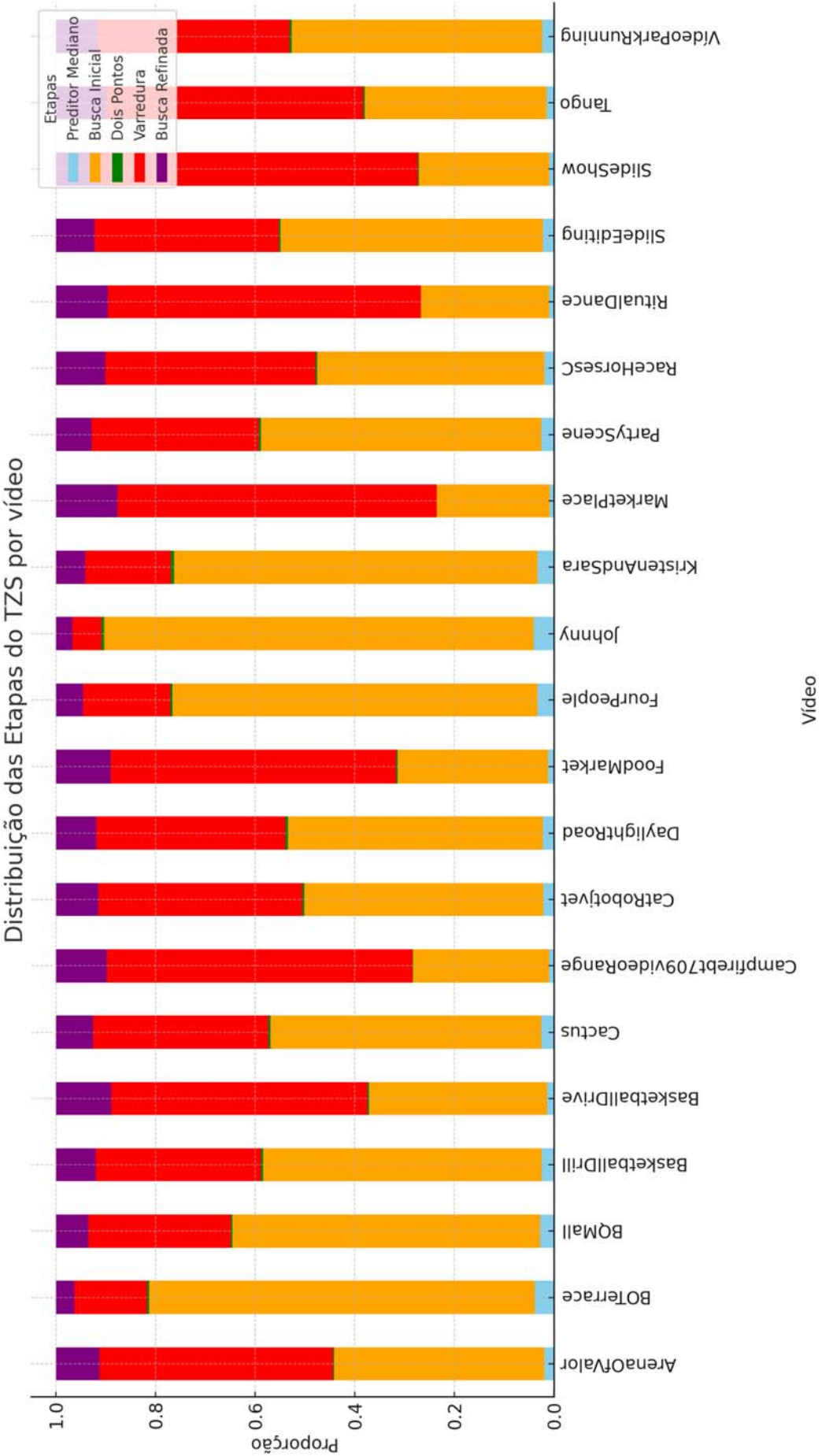


Figura 53 – Distribuição Entre Etapas Tzs Blocos 32x32