

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



Dissertação

**Exploração no Espaço de Projeto de Arquiteturas para os Modos Angulares da
Predição Intra Quadro do Padrão VVC**

Vinicius de Araujo Borges

Pelotas, 2024

Vinicius de Araujo Borges

**Exploração no Espaço de Projeto de Arquiteturas para os Modos Angulares da
Predição Intra Quadro do Padrão VVC**

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Computação

Orientador: Luciano Volcan Agostini
Co-Orientador: Marcelo Schiavon Porto

Pelotas, 2024

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

B732e Borges, Vinicius de Araujo

Exploração no espaço de projeto de arquiteturas para os modos angulares da predição intra quadro do padrão VVC [recurso eletrônico] / Vinicius de Araujo Borges ; Luciano Volcan Agostini, orientador ; Marcelo Schiavon Porto, coorientador. — Pelotas, 2024.

79 f. : il.

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

1. Design de hardware. 2. Predição intra quadro. 3. Modos Angulares. 4. Versatile Video Coding. I. Agostini, Luciano Volcan, orient. II. Porto, Marcelo Schiavon, coorient. III. Título.

CDD 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Vinicius de Araujo Borges

Exploração no Espaço de Projeto de Arquiteturas para os Modos Angulares da
Predição Intra Quadro do Padrão VVC

Dissertação aprovada, como requisito parcial para obtenção do grau de Mestre em
Computação, Programa de Pós-Graduação em Computação, Centro de
Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da defesa: 28 de Agosto de 2024.

Banca examinadora:

Prof. Dr. Luciano Volcan Agostini (Orientador)

Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

Prof. Dr. Guilherme Ribeiro Correa

Doutor em Ciência da Computação pela Universidade de Coimbra

Prof. Dr. Daniel Munari Palomino

Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

Prof. Dr. Marcel Moscarelli Correa

Doutor em Computação pela Universidade Federal de Pelotas

Agradecimentos

Gostaria de começar agradecendo ao governo brasileiro e às agências de fomento que através da bolsa de pesquisa, possibilitaram que eu investisse mais tempo em meus estudos e estivesse hoje completando mais um grau acadêmico. Junto destes, porém não menos importante, vem o povo brasileiro, que com a contribuição de seus impostos, financiaram o desenvolvimento educacional de parte da população onde faço parte.

Gostaria de agradecer ao meu laboratório de pesquisa, ViTech, aos meus companheiros de pesquisa e aos professores. Em especial gostaria de agradecer ao meu orientador, Luciano, que sempre me apoiou durante todos esses anos que trabalhamos juntos. Com uma dedicação incrível, onde mesmo em momentos difíceis tinha tempo para todos. Que via o lado bom das coisas quando eu mesmo não enxergava. Você é uma inspiração para todos à sua volta. Muito obrigado por ser meu orientador e por ser essa pessoa incrível.

Gostaria de agradecer aos meus amigos mais próximos, em especial ao Wendel, Iago, Pedro, Rodrigo, Christopher e Ígor. Durante o período do meu mestrado, várias acontecimentos ocorreram, alguns melhores e outros piores, como a pandemia de COVID-19. Esse momento em minha vida foi marcado por muitos altos e baixos, onde eu poderia dizer que este foi o período mais difícil da minha vida até o momento. Gostaria de dizer à todos que citei acima neste parágrafo que, se tive momentos bons durante esse período, parte deles foram graças à vocês, que mesmo não necessariamente sabendo dos problemas que eu enfrentava, fizeram questão de serem presentes em minha vida, me alegrando e me motivando, cada um do seu jeito. Agradeço de coração por ter vocês em minha vida.

Não posso deixar de agradecer também à minha família. Primeiramente ao meu pai, Zoe Lucas, que me ensinou muito ao longo de minha vida de forma direta ou indireta. Me ensinou valores como o respeito, gratidão, humildade, responsabilidade e tantos outros. Lembro como se fosse ontem, eu criança comendo chocolate, ele em tom de brincadeira me pedia um pedaço e quando eu oferecia, me dizia que era só uma brincadeira, que não queria. Pode ser que para ele fosse só uma brincadeira, mas ali ele me ensinou que não devemos ser egoístas, que devemos compartilhar, e isso são coisas que levo para vida. Se sou quem sou hoje, devo isso à ele.

Agradeço à minha irmã Natália, por estar presente na minha vida. Durante a pandemia de COVID-19, muitas coisas deram errado na minha vida, mas mesmo assim, tive a grata surpresa de poder conversar e conhecer melhor a incrível pessoa que ela é. Compartilhamos ideias e opiniões, dividimos as dores de nossos problemas e enfrentamos nossos medos unidos para que pudéssemos seguir em frente. Minha irmã é com certeza uma das pessoas que me motiva a ser alguém melhor todos os dias. Agradeço de coração por ser minha irmã e uma amiga próxima.

Agradeço profundamente à minha querida mãe Elizabete, meu pilar, minha inspiração, imparável, incansável. Mãe essa que fez das tripas coração para criar eu e minha irmã. Fez de tudo para que não faltasse nada para nós. Trabalhava, cuidava da casa e ainda tinha tempo para todos. Quantos dias e noites passou ao meu lado em minha infância para garantir que eu estudasse e compreendesse melhor a matéria da escola. Sempre zelando por meus estudos, mesmo que estivesse esgotada de um dia cansativo, pois sabia que aquilo era importante para minha formação. Me deu amor, me deu carinho, me ensinou tudo, assim como meu pai. Palavras não bastam para descrever o quão importante ela é para mim. Ela é a famosa uma mãe em um milhão. Mãe, quero que saiba que você é uma das pessoas que mais amo e respeito.

Por fim, e mais importante de todos, gostaria de agradecer do fundo do meu coração a minha parceira, minha vida, minha noiva Giuli. Só ela sabe tudo que eu passei nesse período, o que nós passamos. Sempre presente nos momentos bons e ruins. Que mesmo não estando bem, faz questão de me levantar quando caio. Que me apoia, me aceita como sou, com meus pontos positivos e negativos. Ela é minha principal motivação para levantar todo dia e querer ser alguém melhor. Já se passaram 11 anos que estamos juntos, e se eu posso dizer que o meu eu do presente é melhor que o do passado, devo muito isso a ela. Dona de uma beleza indescritível e de um sorriso exuberante, ela transforma meu mundo cinza em um mundo cheio de cor. Giuli, gostaria de te agradecer por tudo que você é, por tudo que você fez, faz e fará por nós. Quero dizer para você que pode contar comigo para sempre, quero dizer que te amo mais do que tudo e nada mudará isso. Muito obrigado por fazer parte da minha vida.

Resumo

BORGES, Vinicius de Araujo. **Exploração no Espaço de Projeto de Arquiteturas para os Modos Angulares da Predição Intra Quadro do Padrão VVC**. 2024. 79f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2024.

A sociedade atual possui uma alta demanda de vídeos digitais para diversos propósitos da vida cotidiana. Junto com essa demanda, os vídeos digitais passam a ter cada vez mais informações conforme suas resoluções e taxas de quadros por segundo aumentam. Quando consideramos novas tecnologias, como vídeos imersivos, essa quantidade de informação pode ser exponencialmente maior. Para atender essa demanda, são necessários os codificadores de vídeo, responsáveis por comprimir vídeos digitais para tamanhos que possam ser armazenados e/ou transmitidos entre dispositivos com as configurações de mercado atuais. O codificador *Versatile Video Coding* (VVC) é um dos codificadores de vídeo mais atuais e foi desenvolvido para fornecer uma maior eficiência de codificação que os padrões de codificação anteriormente desenvolvidos, além de possibilitar uma maior versatilidade para diversos tipos de vídeos diferentes. Para que seja possível que o padrão codifique vídeos digitais de forma eficiente, o VVC possui diversas inovações em cada uma das ferramentas básicas de um codificador de vídeo. Esta dissertação propõe a criação de duas heurísticas e três arquiteturas de hardware para a ferramenta de modos angulares da predição intra quadro do padrão de codificação VVC. O objetivo deste trabalho é apresentar diferentes estratégias com foco em redução de custo computacional e/ou eficiência de codificação. Ambas heurísticas apresentadas possuem como foco a redução de custo computacional da ferramenta modos angulares utilizando estratégias como redução de modos angulares e tamanhos de bloco disponíveis. Duas arquiteturas criadas são baseadas nas heurísticas propostas, enquanto a terceira arquitetura foi criada com foco em não gerar impacto na eficiência de codificação e, portanto, suporta todos os modos e todos os tamanhos de bloco definidos para a ferramenta no VVC. Os resultados extraídos das heurísticas mostram que as reduções em custo computacional obtidas, considerando todo o codificador VVC, variam de 18,72% a 60,92% de redução no tempo de codificação. Para que fosse possível chegar nesses valores de redução de custo computacional, as heurísticas geraram perdas na eficiência de codificação que variam de 2,17% a 8,62% de aumento no BD-Rate. Nos resultados de síntese, as arquiteturas de hardware desenvolvidas são capazes de codificar vídeos na resolução 1080p@30qps com frequências entre 75,8 MHz e 131,3 MHz. A potência dissipada para essas frequências ficou no intervalo de 91,65 mW e 755,18 mW, enquanto a área necessária para cada arquitetura ficou entre 1.453 k portas e 13.508,4 k portas NAND2.

Palavras-chave: Design de hardware; Predição intra quadro; Modos Angulares; *Versatile Video Coding*.

Abstract

BORGES, Vinicius de Araujo. **Design Space Exploration of Architectures for Angular Modes of Intra-Frame Prediction of the VVC Pattern**. 2024. 79p. Dissertation (Master degree in Computer Science) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2024.

Today's society has a high demand for digital videos for various purposes in everyday life. Along with this demand, digital videos are increasingly having more information as their resolutions and frame rates per second increase. When we consider new technologies such as immersive videos, this amount of information can be exponentially greater. To meet this demand, video encoders are needed, responsible for compressing digital videos to sizes that can be stored and/or transmitted between devices with current market configurations. The Versatile Video Coding (VVC) encoder is one of the most current video encoders and was developed to provide greater coding efficiency than previously developed coding standards, in addition to enabling greater versatility for several different types of videos. In order for the standard to encode digital videos efficiently, VVC has several innovations in each of the basic tools of a video encoder. This dissertation proposes the creation of two heuristics and three hardware architectures for the angular modes tool of intra-frame prediction of the VVC coding standard. The objective of this work is to present different strategies focused on reducing computational cost and/or coding efficiency. Both heuristics presented focus on reducing the computational cost of the angular modes tool using strategies such as reducing angular modes and available block sizes. Two architectures created are based on the proposed heuristics, while the third architecture was created with a focus on not generating an impact on coding efficiency and, therefore, supports all modes and all block sizes defined for the tool in VVC. The results extracted from the heuristics show that the reductions in computational cost obtained, considering the entire VVC encoder, range from 18.72% to 60.92% of encoder time reduction. In order to reach these computational cost reduction values, the heuristics generated losses in coding efficiency that range from 2.17% to 8.62% of BD-Rate increase. In the synthesis results, the developed hardware architectures are capable of encoding videos at 1080p@30fps resolution with frequencies between 75.8 MHz and 131.3 MHz. The power dissipated for these frequencies was in the range of 91.65 mW and 755.18 mW, while the required area for each architecture was between 1,453 k gates and 13,508.4 k NAND2 gates.

Keywords: Hardware Design; Intra-frame Prediction; Angular Modes; Versatile Video Coding.

Lista de Figuras

Figura 1 - Modelo Genérico de um codificador de vídeo.....	15
Figura 2 - Tipos de partições do sistema de particionamento QTMT	18
Figura 3 - Estrutura de particionamento de blocos QTMT do VVC	19
Figura 4 - Modos angulares da predição intra quadro do VVC. Fonte: (CHEN; YE; KIM, 2020)	21
Figura 5 - Processo do modo DC da predição intra quadro	22
Figura 6 - Processo do modo Planar da predição intra quadro	24
Figura 7 - Processo dos modos angulares da predição intra quadro. Exemplo do modo angular 14	25
Figura 8 - Ferramenta de múltiplas linhas de referência da predição intra quadro do padrão de codificação VVC	28
Figura 9 - Ferramenta de predição intra quadro baseada em matriz do padrão de codificação VVC	29
Figura 10 - Ferramenta de subpartição intra quadro do padrão de codificação VVC	30
Figura 11 - Porcentagem de uso dos modos angulares no VTM	38
Figura 12 - Porcentagem de uso dos tamanhos de bloco para os modos angulares utilizando os QPs 22 e 37 no VTM	40
Figura 13 - Diagrama parte topo da arquitetura Sem Redução de Custo Computacional (SRCC) proposta para a predição intra quadro do VVC.....	53
Figura 14 - Diagrama parte topo da arquitetura de Redução Forte de Custo Computacional (RFCC) proposta para a predição intra quadro do VVC	55
Figura 15 - Diagrama parte topo da arquitetura de Redução Balanceada de Custo Computacional (RBCC) proposta para a predição intra quadro do VVC.....	57
Figura 16 - Diagrama de bloco do modo Planar.....	58
Figura 17 - Diagrama de blocos do modo DC	61
Figura 18 - Diagrama de blocos dos modos angulares, utilizando como exemplo o modo angular 44	62

Lista de Tabelas

Tabela 1 - Fator angular dos modos angulares da predição intra quadro do VVC, onde MA representa modo angular e FA representa fator angular	27
Tabela 2 - Média de BD-Rate e RTC por classe de sequências de vídeo do CTC para heurística RFCC.....	42
Tabela 3 - Análise de BD-Rate sobre redução de modos angulares para a predição intra quadro	44
Tabela 4 - Análise de BD-Rate sobre redução de tamanhos de bloco para a predição intra quadro	46
Tabela 5 - Média de BD-Rate e RTC por classe de sequências de vídeo do CTC para heurística RBCC.....	50
Tabela 6 - Tabela de ciclos de clock para cada CU e para cada CTU das arquiteturas RFCC, SRCC e RBCC	65
Tabela 7 - Resultados de síntese.....	66
Tabela 8 - Comparações com trabalhos relacionados	68

Lista de Abreviaturas e Siglas

AI	<i>All-Intra</i>
AIP	<i>Angular Intra Prediction</i>
BD-Rate	<i>Bjontegaard Delta Rate</i>
Bit	Menor unidade de informação que pode ser armazenada
BRAM	<i>Block Random Access Memory</i>
BT	<i>Binary Tree</i>
CABAC	<i>Context-based Adaptive Binary Arithmetic Coding</i>
Cb	<i>Blue Chrominance</i>
CB	<i>Coding Blocks</i>
Cr	<i>Red Chrominance</i>
CTC	<i>Common Test Conditions</i>
CTU	<i>Coding Tree Unit</i>
CU	<i>Coding unit</i>
DSP	<i>Digital Signal Processing</i>
FHD	<i>Full High Definition</i>
FPS	<i>frames per second</i>
HD	<i>High Definition</i>
HEVC	<i>High-Efficiency Video Coding</i>
HM	<i>HEVC Test Model</i>
ISP	<i>Intra Sub-Partition</i>
JVET	<i>Joint Video Experts Team</i>
LUT	<i>Lookup Table</i>
MIP	<i>Matrix-based Intra Prediction</i>
MPEG	<i>Moving Picture Experts Group</i>
MPM	<i>Most Probable Mode</i>
MRL	<i>Multiple Reference Line Prediction</i>
MTT	<i>Multi-type Tree</i>
PU	<i>Prediction Unit</i>
QP	<i>Quantization Parameter</i>
QPS	Quadros por segundo

QT	<i>Quadtree</i>
QTMT	<i>Quadtree with Nested Multi-Type Tree</i>
RBCC	Redução Balanceada de Custo Computacional
RD-list	<i>Rate-Distortion list</i>
RDO	<i>Rate Distortion Optimization</i>
RFCC	Redução Forte de Custo Computacional
RGB	<i>Red, Green, and Blue</i>
RMD	<i>Rough Mode Decision</i>
RTC	Redução de tempo de codificação
SRCC	Sem Redução de Custo Computacional
TT	<i>Ternary Tree</i>
UHD	<i>Ultra-High Definition</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VTM	<i>VVC Test Model</i>
VVC	<i>Versatile Video Coding</i>
Y	<i>Luminance</i>

Sumário

1 Introdução	9
2 Conceitos Básicos	12
2.1 Particionamento de Blocos no VVC	17
2.2 Predição Intra Quadro do VVC	20
3 Trabalhos Relacionados	32
4 Heurísticas Desenvolvidas	36
4.1 Configuração Experimental	36
4.2 Avaliações Preliminares	37
4.3 Heurística de Redução Forte de Custo Computacional	40
4.5 Heurística de Redução Balanceada de Custo Computacional	43
5 Implementações de Hardware	51
5.1 Parte Topo das Arquiteturas	51
5.1.1 Arquitetura SRCC	52
5.1.2 Arquitetura RFCC	54
5.1.3 Arquitetura RBCC	56
5.2 Unidades de Processamento dos Modos Angulares	58
5.2.1 Unidade Planar	58
5.2.2 Unidade DC	60
5.2.3 Unidades Direcionais	61
6 Resultados e Discussão	64
7 Considerações Finais	73
Referências	77

1 Introdução

Nos últimos anos houve um aumento significativo na demanda por transmissão de vídeos digitais na internet por parte da sociedade para diversas finalidades. Esta demanda aumentou ainda mais por causa da pandemia de COVID-19. Com a pandemia, novas formas de estudo e trabalho foram exploradas pela população, como reuniões virtuais, *home office* e escolas com mais tarefas para serem realizadas de forma remota. Tudo isso levou a mais pesquisas sobre o processo de codificação de vídeo (PALAU et al., 2021), que é o processo referente à compressão dos dados que compõem vídeos digitais.

Codificar um vídeo exige muito tempo, esforço computacional e consumo de energia, devido às diversas etapas necessárias para serem executadas por cada codificador. Essas etapas incluem predição inter quadros, predição intra quadro, transformadas direta e inversa, quantização direta e inversa, filtro de laço e codificador de entropia (AGOSTINI, 2007).

O atual codificador de vídeo estado da arte é o *Versatile Video Coding* (VVC) (CHEN et al., 2023), desenvolvido pelo *Joint Video Experts Team* (JVET), do qual o *ISO/IEC Moving Picture Experts Group* (MPEG) fez parte, e disponibilizado em julho de 2020. Apesar dos requisitos computacionais significativamente elevados, o VVC oferece taxas de compressão notáveis e supera outros codificadores disponíveis comercialmente em termos de eficiência de compressão (BROSS et al., 2021).

Este trabalho está focado na predição intra quadro, que visa reduzir a redundância espacial de um quadro. Redundância espacial é a tendência que as amostras de um dado quadro têm de possuir dados muito semelhantes às suas amostras vizinhas (geralmente são utilizadas amostras vizinhas já preditas à esquerda e acima). A predição intra quadro utiliza essas amostras vizinhas como base para codificar uma determinada amostra de um quadro (SALDANHA et al., 2021). O VVC possui diversas ferramentas importantes para a predição intra quadro, como modos angulares (CHEN; YE; KIM, 2020), múltiplas linhas de referência (do inglês *Multiple Reference Line* - MRL) (BROSS, 2018), predição intra baseada em matriz (do inglês *Matrix-based Intra Prediction* - MIP) (SALDANHA et al., 2021) e subpartição intra (do inglês *Intra Subpartition* - ISP) (SALDANHA et al., 2021).

A ferramenta modos angulares, foco deste trabalho, é uma ferramenta herdada dos codificadores anteriores (como o *High-Efficiency Video Coding* - HEVC) e codifica a amostra a ser predita com as amostras vizinhas baseado no ângulo do modo. Estão juntos nessa ferramenta os modos Planar e DC, que não são angulares propriamente, devido seu processamento ser realizado de forma diferente dos modos angulares, mas que foram agrupados a esta ferramenta. No VVC, são 67 modos angulares ao todo. 35 desses 67 modos são herdados do HEVC (Planar, DC e 33 modos angulares), enquanto que os outros 32 modos angulares foram adicionados no VVC (BROSS et al., 2020).

O codificador VVC, apesar de possuir uma maior eficiência de codificação em comparação com seu antecessor HEVC, também possui um custo computacional significativamente mais elevado. Esse custo computacional a mais pode resultar em grandes impactos na implementação de arquiteturas de hardware para o codificador VVC. Pensando em entender melhor o impacto que as melhorias realizadas na ferramenta podem ter à nível de arquiteturas de hardware, este trabalho apresentará três propostas de arquitetura para a ferramenta modos angulares da predição intra quadro do codificador VVC. Essas arquiteturas possuem focos distintos e duas delas são baseadas em heurísticas desenvolvidas através de análise do comportamento da ferramenta de modos angulares do VVC. Essa análise foi realizada no software de referência do VVC (VTM) (CHEN; YE; KIM, 2020).

Essa dissertação começa apresentando conceitos básicos da área e o padrão de codificação foco desse trabalho, o VVC. Dentro desse capítulo são brevemente discutidos os tópicos que envolvem esse trabalho em relação ao VVC: o particionamento de blocos e a predição intra quadro. Dentro da predição intra quadro, todos os modos de predição presentes serão apresentados. Como este trabalho é baseado na ferramenta modos angulares da predição intra quadro, essa ferramenta terá uma explicação mais detalhada no texto. A seguir, trabalhos existentes na literatura atual que tem relação com essa dissertação serão brevemente apresentados.

O capítulo de heurísticas desenvolvidas apresenta o método experimental do presente trabalho sobre a parte de software. Dentro deste capítulo, é discutido a configuração experimental utilizado para realizar experimentos no software de referência VTM (CHEN; YE; KIM, 2020). Junto à configuração, são apresentados os

resultados preliminares de testes realizados no VTM. Por fim, duas subseções apresentam as duas heurísticas propostas neste trabalho. Estas heurísticas são apresentadas, juntamente com os resultados da sua avaliação em relação ao VTM.

No capítulo de implementações de hardware são apresentadas as três arquiteturas implementadas neste trabalho. Cada arquitetura possui sua própria seção, onde são discutidas suas implementações.

A seguir, são apresentados os resultados dessas arquiteturas no capítulo de resultados e discussão. Esses resultados serão comparados com os trabalhos encontrados na literatura que foram descritos na seção de trabalhos relacionados. Por fim, essa dissertação apresentará um capítulo com as considerações finais sobre o trabalho.

2 Conceitos Básicos

Nessa seção será realizada uma revisão dos conceitos básicos relacionados a vídeos digitais. Após, será apresentado o padrão de codificação de vídeo VVC, bem como algumas de suas características relevantes para esse trabalho, analisando mais a fundo as ferramentas do codificador trabalhadas nesse projeto.

Grande parte da população, quando assiste a um vídeo digital, não imagina que um vídeo é, na verdade, um conjunto de imagens estáticas. Essas imagens, chamadas de quadros, são matrizes de duas dimensões de píxeis. Um pixel é a menor unidade com informação de luz e cor em uma imagem. Cada quadro é formado por milhares de píxeis. O tamanho desse quadro, também chamado de resolução, pode variar bastante (WIEN, 2015). Atualmente existem diversos tamanhos de resolução disponíveis. Algumas resoluções são bastante utilizadas no atual momento pela população, como as resoluções FHD (*Full-High Definition*) e UHD (*Ultra-High Definition*), presentes em plataformas de *streaming* por exemplo. O tamanho da resolução FHD é 1920x1080 pixels, onde 1920 é o tamanho horizontal da resolução, enquanto 1080 é o tamanho vertical da resolução. O tamanho da resolução UHD é de 3840x2160 píxeis, que é chamada de UHD 4K. Já existem aplicações com resoluções UHD 8K (como por exemplo televisões), com 7680x4320 píxeis. A necessidade de resoluções cada vez maiores vem do fato que quanto mais píxeis em uma imagem, mais detalhes serão representados por esses píxeis e, conseqüentemente, melhor será a qualidade dessa imagem. Ou seja, quanto maior a resolução, e principalmente quanto maior a taxa de bits, melhor tende a ser a qualidade visual.

Mas para entender melhor os vídeos digitais, é necessário primeiro entender como o ser humano recebe informações de imagens. O sistema visual humano possui células da retina que agem como fotorreceptores chamadas de bastonetes e cones que são sensíveis à luz. Os bastonetes são utilizados para capturar a luminosidade da imagem, enquanto os cones são sensíveis às cores azul, vermelho e verde (POLLACK, 2006). O sistema visual humano é capaz de perceber milhares de cores distintas a partir de combinações de intensidades diferentes sobre as cores perceptíveis aos cones. Além disso, o sistema visual humano é capaz de distinguir mais de 60 tons de cinza, que indicam a intensidade de luminosidade da imagem (luminância) (GONZALEZ; WOODS, 2009).

Espaço de cor é a definição utilizada para representar um sistema que representa cores em um vídeo digital. O pixel geralmente é formado por três amostras de um dado espaço de cor. Existem alguns tipos de espaços de cores utilizados em imagens digitais, como por exemplo: vermelho, verde e azul (do inglês *red, green and blue* - RGB) e YCbCr (SHI; SUN, 2019). O espaço de cor RGB representa as cores primárias vermelho, verde e azul em três matrizes diferentes. Diferente do RGB, o espaço de cores YCbCr, comumente utilizado nos padrões de codificação atuais, apresenta por sua vez três componentes: a luminância (Y), a cromaância azul (Cb) e a cromaância vermelha (Cr) (BHASKARAN; KONSTANTINIDES, 1997). O principal motivo da utilização do YCbCr se deve pelo fato do sistema visual humano ser mais sensível à luminosidade do que à cor e, separando a informação de luminância da informação de cor, é possível aplicar técnicas distintas de compressão em ambas as informações, ampliando as taxas de compressão sem prejudicar a qualidade da imagem.

Cada codificador suporta diferentes configurações de subamostragem de espaço de cor, como por exemplo a configuração de subamostragem 4:2:2. A subamostragem é um método de compressão de um sinal digital onde é feita a manipulação da informação de cor. Essa manipulação visa reduzir a quantidade de dados usados para representar a informação de cor e, assim, ampliar as taxas de compressão, visto que o sistema visual humano é menos sensível à cor. O exemplo de configuração de subamostragem 4:2:2, citado acima, significa que, para cada quatro amostras de Y, terão duas amostras de Cb e duas amostras de Cr. O padrão de codificação de vídeo VVC utiliza as configurações de subamostragem 4:2:2, 4:2:0, 4:4:4 e 4:0:0 (SALDANHA et al., 2021).

Também é importante entender como uma imagem digital se transforma em um vídeo. A resposta surge quando uma sequência de imagens digitais, muito semelhantes, são apresentadas ao usuário em um certo período de tempo. Uma pessoa começa a sentir a sensação de movimento mais suave a partir de vinte e quatro quadros por segundo (qps), ou *frames per second* em inglês (fps) (WIEN, 2015). Nos dias atuais, existem aplicações com vídeos digitais que podem utilizar 60 qps ou 120 qps, dependendo da aplicação até mais qps são necessários, como tecnologias de vídeos imersivos, realidade virtual ou aumentada (SALMON; ARMSTRONG; JOLLY, 2011) e até mesmo *ultra slow motion*.

Um quadro de tamanho 1920x1080 possui mais de dois milhões de píxeis, onde cada pixel possui três amostras de cor, com informações de Y, Cb e Cr. Esse valor é considerado para apenas um quadro, sendo que os vídeos usados nos dias atuais utilizam pelo menos trinta quadros por segundo para geração do vídeo digital. Assim, um filme de duas horas teria 216.000 quadros, ou 447.897.600.000 píxeis, ou 1.343,7 GB, caso uma amostra ocupe um byte. Então, é perceptível a quantidade gigantesca de dados que um vídeo digital pode ter na atualidade. Com todos esses dados, seria impossível armazenar ou transmitir vídeos deste tamanho.

Para solucionar esse problema, os padrões de codificação de vídeo foram criados. Esses padrões têm como objetivo reduzir drasticamente a quantidade de dados redundantes presentes nas informações de um vídeo digital. Essas redundâncias presentes em um vídeo podem ser: espaciais, temporais e/ou entrópicas (GHANBARI, 2003; SHI; SUN, 2019). Redundância espacial é quando amostras de um mesmo quadro possuem uma mesma informação. Para remover a redundância espacial, os padrões de codificação utilizam a predição intra quadro (AGOSTINI, 2007), as transformadas e quantização (AGOSTINI, 2007). Redundância temporal é quando amostras de quadros diferentes (quadros vizinhos) possuem a mesma informação. Para evitar esse tipo de redundância, os padrões usam a predição inter quadros (AGOSTINI, 2007). Por fim, redundância entrópica refere-se à probabilidade de ocorrência dos símbolos, onde alguns símbolos aparecem com uma frequência maior que outros símbolos. É importante ressaltar que as etapas de predição, ao explorarem as redundâncias espaciais e temporais, tem como objetivo aumentar a redundância entrópica para que o codificador de entropia possa realizar a compressão de fato. A codificação entrópica, para reduzir a redundância entrópica, define menos quantidade de bits para representar símbolos que possuem uma maior probabilidade de ocorrência, enquanto define mais quantidade de bits para representar símbolos menos frequentes (AGOSTINI, 2007).

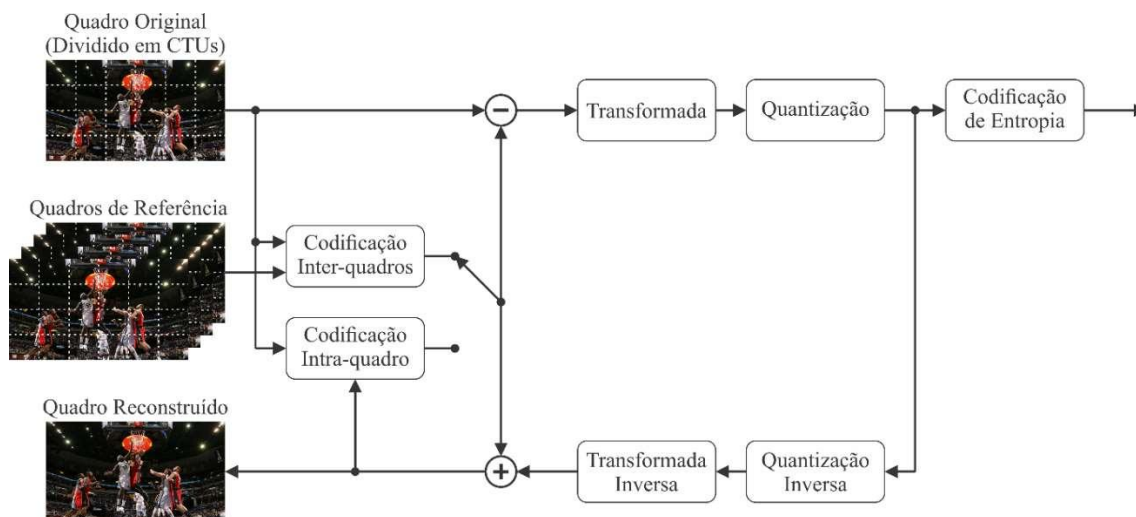
O padrão de codificação de vídeos VVC, também conhecido como H.266 foi desenvolvido com dois principais objetivos: melhoria na performance de compressão e amplo suporte para diversas aplicações. O VVC foi desenvolvido pelo grupo *Joint Video Experts Team* (JVET) (“JVET - Joint Video Experts Team”, [s.d.]) e teve sua primeira versão publicada em julho de 2020. Quando comparado com seu antecessor, o HEVC, o novo padrão de codificação possui uma taxa de compressão cerca de 50%

melhor (BROSS et al., 2021). Para que fosse possível alcançar essa taxa de compressão em relação ao padrão de codificação anterior, novas ferramentas foram criadas para o VVC, além de melhorias em ferramentas herdadas de seu antecessor, o HEVC.

Nesse momento, é importante entendermos como é o funcionamento de um codificador de vídeo. Para isso, um modelo genérico de codificador de vídeo é apresentado pela Figura 1. O processo de codificação começa com o quadro atual de entrada (quadro original) que é o quadro que será processado. Esse quadro é enviado para a predição inter quadros e a predição intra quadro.

A predição inter quadros explora as redundâncias temporais de quadros de um dado vídeo. Essa predição utiliza como entrada quadros de referência, que são quadros anteriormente codificados, além do quadro original que será processado, como mostra a Figura 1. Nesse sentido, o quadro original atual, quando codificado, será utilizado como quadro de referência para os próximos quadros originais. O processo de como ele será reutilizado será explicado mais a frente.

Figura 1 - Modelo Genérico de um codificador de vídeo



Fonte: Elaborada pelo autor (2024).

A predição intra quadro explora as redundâncias espaciais de um quadro de um vídeo digital. Para isso, essa predição utiliza apenas o quadro atual como entrada. Na Figura 1, é possível perceber que a predição intra quadro também recebe como entrada o quadro atual reconstruído para realizar sua predição. Assim como na predição inter quadros, a processo de como o quadro atual será utilizado após ser

codificado será explicado conforme avançamos no fluxo do modelo genérico de codificador de vídeo.

Os blocos do quadro original atual serão codificados com a predição inter quadros ou predição intra quadro. A decisão de qual modo deve ser utilizado para cada bloco é tomada pelo decisor de modo, a qual é representada pela chave seletora presente na Figura 1. Após um bloco do quadro original atual ser codificado por uma das predições disponíveis, os valores gerados desse bloco processado são subtraídos dos valores contidos no bloco original, gerando o que chamamos de resíduo.

Esse resíduo é enviado para os próximos módulos que serão responsáveis por reduzir a redundância espacial deste no domínio das frequências (transformada e quantização). A transformada tem como função transformar a informação presente no resíduo que está no domínio espacial para o domínio das frequências. Após, a informação do resíduo, já no domínio das frequências, é enviada para quantização que será responsável por reduzir as redundâncias presentes nesse domínio, como mostra o fluxo na Figura 1. Por fim, o resultado da quantização é enviado para a codificação de entropia, que realiza reduções na parte de redundâncias entrópicas (forma como os dados são codificados), gerando a saída que é o *bitstream* codificado.

O *bitstream* codificado é o resultado da codificação de um dado vídeo, possibilitando que este vídeo seja armazenado em dispositivos ou transmitido entre os mesmos. Para que seja possível visualizar o vídeo novamente, esse *bitstream* necessita ser reconstruído por um decodificador. Porém, o bitstream que será a informação recebida (o decodificador não recebe o quadro original) pelo decodificador, é o resultado de uma codificação de um vídeo original e, como a codificação gera perdas de informação, o quadro codificado será diferente do quadro original após a decodificação (RICHARDSON, 2003). Para solucionar esse problema, o codificador passa a descartar o quadro original, após o mesmo ser processado, e armazena apenas o quadro reconstruído (como pode ser visto na Figura 1). Com isso, o codificador passa a utilizar os quadros processados como referência, assim como o decodificador.

O processo de reconstrução de um quadro já processado começa após a parte de quantização, pois a parte de possíveis perdas de informação entre um quadro original e um quadro processado vai até a quantização (a codificação e decodificação de entropia não geram perdas de informação). Como representado na Figura 1, é

aplicado a quantização inversa e a transformada inversa, gerando os resíduos reconstruídos. Esses resíduos são somados ao resultado gerado pela codificação inter quadros ou intra quadro (dependendo da decisão tomada pelo hardware de controle do codificador), gerando o bloco reconstruído. O bloco reconstruído é agora armazenado para ser utilizado como referência para processamento dos próximos blocos tanto para a predição inter quadros, quanto para predição intra quadro.

É importante ressaltar que esse processo representado pela Figura 1 é um diagrama simples do fluxo de processamento de um codificador de vídeo. Os atuais codificadores de vídeo, como o VVC, possuem diversas ferramentas dentro de cada uma das partes descritas logo acima para realizar suas respectivas funções. Como este trabalho propõe soluções para uma das partes citadas pela Figura 1, o resto dessa seção será destinada a apresentação, de forma mais detalhada, das ferramentas presentes na parte de interesse a este trabalho.

2.1 Particionamento de Blocos no VVC

O particionamento de blocos realiza a divisão de um quadro em diversos blocos de tamanho definido pelo padrão de codificação, a fim de desempenhar uma melhor codificação desse quadro. Com a divisão de quadros em blocos menores é possível utilizar diversos métodos diferentes de compressão em um mesmo quadro. Isso possibilita a utilização do melhor método de compressão para cada região específica de um dado quadro.

No VVC, os quadros do vídeo a serem codificados são primeiramente particionados em blocos, denominados unidades de codificação em árvore (do inglês *Coding Tree Units* - CTU), contendo um número de píxeis que varia de 32×32 a 128×128 . Atuando como a raiz de uma estrutura de árvore, cada CTU é subdividida em unidades de codificação (do inglês *Coding Units* - CU). As CUs contêm as três informações de cores dos píxeis e, quando as informações de cores são separadas, os blocos resultantes são chamados de blocos de codificação (do inglês *Coding Blocks* - CB). Assim, uma CU é composta por três CBs. As CBs são as unidades básicas de processamento das ferramentas de codificação.

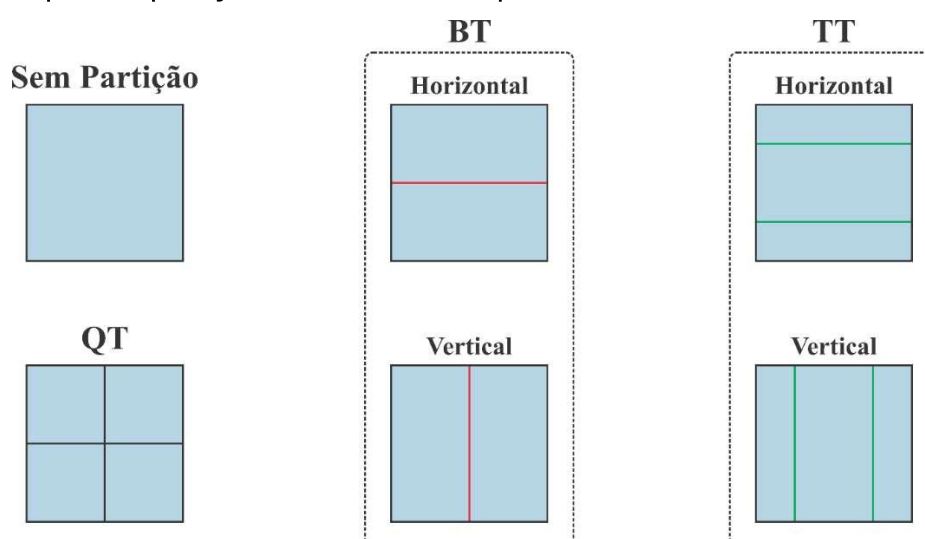
O VVC, assim como seu antecessor HEVC, utiliza a partição de blocos baseado em árvores de codificação. O VVC, além de herdar a estrutura de partição de árvores

quaternárias (do inglês *Quadtree* - QT) do HEVC, introduz a estrutura de particionamento de árvores de múltiplos tipos (do inglês *Multi-Type Tree* - MTT). A estrutura MTT permite o uso de blocos retangulares através de árvores binárias (do inglês *Binary Tree* - BT) e árvores ternárias (do inglês *Ternary Tree* - TT). A junção das estruturas QT e MTT forma a estrutura de particionamento do VVC, chamada de árvores quaternárias aninhadas com árvores de múltiplos tipos (do inglês *Quadtree with Nested Multi-Type Tree* - QTMT).

Como pode ser visto na Figura 2, a estrutura QTMT permite seis tipos diferentes de partições. O primeiro tipo de partição representado na Figura 2 é a codificação sem partição, ou seja, a CU será processada com seu tamanho atual, sem a necessidade de novas partições. O segundo tipo é a partição usando o sistema QT, onde a CU é dividida em quatro CUs quadradas de mesmo tamanho. Os quatro últimos tipos são relativos ao sistema MTT, utilizando árvores binárias BT e árvores ternárias TT.

Como mostra a Figura 2, o sistema BT particiona uma CU no meio, gerando duas novas CUs simétricas. A partição BT pode ser realizada tanto da forma vertical quanto da forma horizontal, segundo a Figura 2. Já a partição TT, divide a CU de uma forma diferente das outras partições com foco em divisões de mesmo tamanho. A partição TT divide a CU em três partes com uma relação de 25%/50%/25% da área da CU, ou seja, das três divisões feitas pela partição TT, a CU do meio terá o dobro do tamanho das outras CUs. Assim como a partição BT, a partição TT também pode ser realizada de forma vertical ou horizontal, como está representado na Figura 2.

Figura 2 - Tipos de partições do sistema de particionamento QTMT



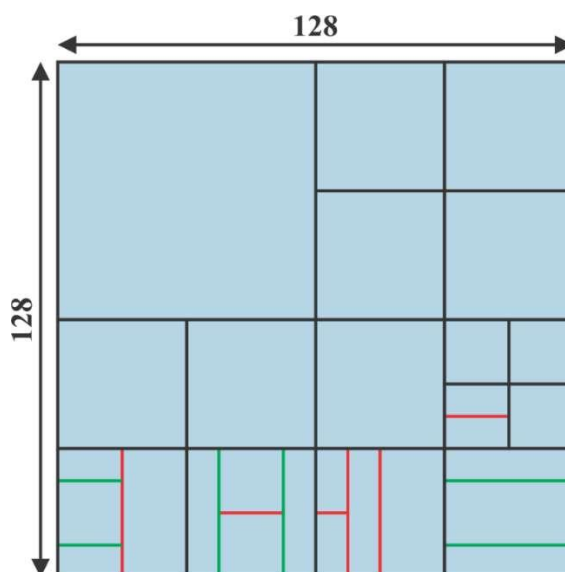
Fonte: Elaborada pelo autor (2024).

A QTMT primeiramente divide blocos maiores (CTU) em blocos quadrados menores (CU) de forma recursiva. Os blocos quadrados que são folha da primeira divisão podem ser divididos adicionalmente em blocos menores não quadrados, sendo raízes de uma nova árvore de tamanho múltiplo, com partições binárias ou ternárias. Uma vez que um bloco quadrado é dividido em blocos não quadrados menores, as próximas divisões não poderão mais ser quaternárias.

A Figura 3 mostra esse processo de partição QTMT do VVC para uma CTU de tamanho 128x128. Dentro da Figura 3 é possível ver pelo menos um exemplo de todos os tipos de partição citados acima e representados pela Figura 2. As linhas em preto representam as divisões do sistema QT sendo utilizadas na CTU. As linhas em vermelho e verde na Figura 3 representam o sistema de particionamento MTT, onde o vermelho é utilizado para partições BT, enquanto o verde é utilizado para partições TT. Como comentado acima e representado na Figura 3, a CTU pode ser primeiramente dividida em CUs pelo sistema QT, porém uma vez que a partição é realizada pela estrutura MTT, essas novas CUs não podem ser novamente particionadas com o sistema QT.

Para predição intra quadro, foco deste trabalho, o VVC utiliza uma CTU máxima de 64x64. São dezessete tamanhos de bloco disponíveis para a predição intra quadro (4x4, 4x8, 8x4, 4x16, 16x4, 8x8, 8x16, 16x8, 4x32, 32x4, 8x32, 32x8, 16x16, 16x32, 32x16, 32x32 e 64x64).

Figura 3 - Estrutura de particionamento de blocos QTMT do VVC



Fonte: Elaborada pelo autor (2024).

2.2 Predição Intra Quadro do VVC

Como comentado anteriormente, um vídeo digital possui uma grande quantidade de redundância espacial, onde dentro de um mesmo quadro, diversas amostras vizinhas possuem valores idênticos ou muito próximos entre si. Tendo isso em vista, a predição intra quadro do padrão de codificação VVC tem como objetivo reduzir a redundância espacial de um vídeo digital. Para isso, a mesma possui algumas ferramentas disponíveis para esse propósito. Dentre elas estão a ferramenta *Angular Intra Prediction* (AIP), *Multiple Reference Line* (MRL), *Matrix-based Intra Prediction* (MIP) e *Intra Subpartition* (ISP).

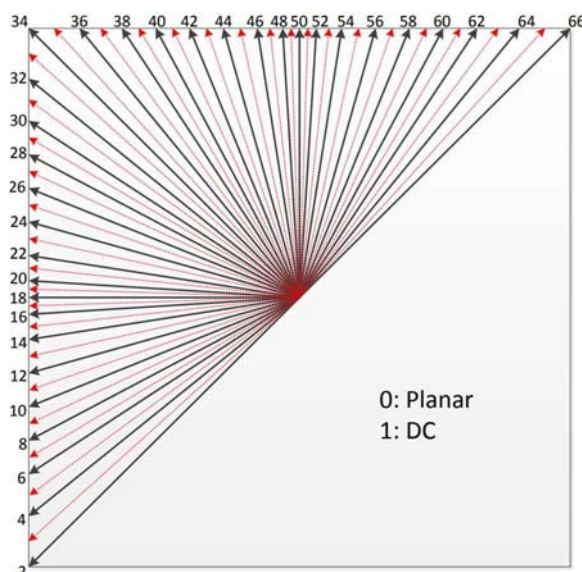
Como existem muitos modos de predição disponíveis, a implementação da predição intra quadro no VTM inclui um recurso presente na norma do VVC chamado de modos mais prováveis (do inglês *Most Probable Modes* - MPM) (PFAFF et al., 2021), onde a representação do modo de predição no fluxo de bits depende dos MPMs, e a heurística de decisão bruta de modos (do inglês *Rough Mode Decision* - RMD) (PFAFF et al., 2021). Com essas ferramentas é possível criar uma lista de candidatos promissores, chamada de lista de taxa-distorção (do inglês *Rate-Distortion list* - RD-list).

A heurística RMD utiliza uma avaliação local para estimar o custo de codificação de cada modo candidato através do custo da taxa-distorção, sem precisar utilizar a otimização de taxa-distorção (do inglês *Rate-Distortion Optimization* - RDO) completa (SULLIVAN; WIEGAND, 1998), que exigiria a codificação completa de cada bloco com cada modo de predição para definir a distorção e a taxa de bits geradas. Com uma decisão local, mesmo que com algum nível de imprecisão, é possível reduzir muito o custo computacional da predição intra quadro do VVC, pois apenas as combinações consideradas com maior potencial serão incluídas na RD-list e serão avaliadas através do RDO completo. A heurística MPM, presente no VVC, seleciona os modos mais prováveis para serem avaliados, com base nos modos usados nos blocos vizinhos e nos modos estatisticamente mais relevantes, e inclui na RD-list. O número de modos na RD-list é dinâmico e depende do tamanho de bloco e do contexto do codificador (PFAFF et al., 2021).

A ferramenta AIP do padrão VVC é foco deste trabalho. Ela possui 67 modos disponíveis para codificação de blocos na parte da predição intra quadro. Desses 67

modos disponíveis, dois deles são modos de predição não-angulares, porém ainda são considerados dentro da ferramenta AIP, como foi citado anteriormente. Esses dois modos são chamados de modo Planar (ou modo 0) e modo DC (ou modo 1), o restante dos 65 modos são todos modos angulares, como pode ser visto na Figura 4.

Figura 4 - Modos angulares da predição intra quadro do VVC. Fonte: (CHEN; YE; KIM, 2020)



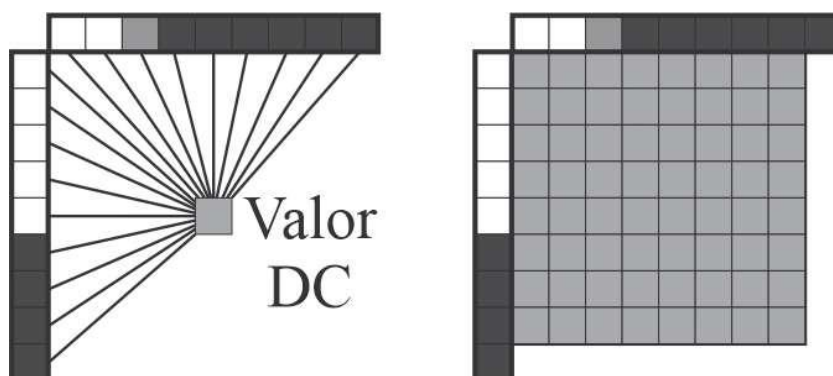
Fonte: Elaborada pelo autor (2024).

Dentro do software de referência do VVC, o VTM, a ferramenta AIP é dividida em AIP-1 e AIP-2. Inicialmente é aplicada a AIP-1 e, com os resultados obtidos, a AIP-2 é então executada. Isso é feito para evitar a avaliação exaustiva de todos os 67 modos disponíveis para cada um dos 17 tamanhos de blocos usados na predição intra quadro do VVC. O AIP-1 avalia todos os modos angulares que o VVC herdou do padrão HEVC. Os modos herdados foram o Planar, o DC e 33 modos angulares (totalizando 35 dos 67 modos utilizados pelo VVC). Esses modos estão representados nas linhas em cinza na Figura 4 (modos 0, 1 e todos os modos pares). A AIP-2 avalia todos os novos modos angulares criados no VVC. Estes são modos intermediários entre cada dois modos angulares da ferramenta AIP-1 (sem contar os modos Planar e DC), totalizando 32 modos dos 67 disponíveis no VVC. Estes modos estão representados pelas linhas vermelhas pontilhadas na Figura 4 (modos ímpares). Com a definição do melhor modo na AIP-1, apenas os modos próximos do escolhido na AIP-1 são avaliados na AIP-2, para reduzir a complexidade do processo.

Quando comparado com o HEVC, é perceptível o aumento significativo no número de modos angulares disponibilizados pelo padrão VVC. O objetivo desse grande aumento tanto nos modos angulares, quanto nos tamanhos de bloco disponíveis para a predição intra quadro é prover a essa predição uma melhor precisão para diferentes padrões de textura.

O modo DC, como mostra a parte esquerda da Figura 5, realiza uma média simples de todas as amostras vizinhas disponíveis à esquerda e acima do bloco a ser predito. Com o resultado dessa média simples é gerado o valor DC. Esse valor é replicado para todas as amostras do bloco a ser predito, como está representado na parte direita da Figura 5. O modo DC é muito eficiente para codificar áreas mais homogêneas do quadro, onde os valores de cada amostra vizinha são muito próximos. É importante ressaltar que as diferentes cores nas amostras vizinhas representam diferentes valores que essas amostras possam ter.

Figura 5 - Processo do modo DC da predição intra quadro



Fonte: Elaborada pelo autor (2024).

O cálculo realizado pelo modo DC, pode ser visto na Equação (1). A Equação (1) mostra o cálculo de predição de uma amostra do bloco na posição $[x][y]$, onde o *ValorDC* é o valor da média de todas as amostras vizinhas utilizadas para a predição do bloco. Na Equação (1), $x = 0..M - 1$ e $y = 0..N - 1$, onde M representa a largura do bloco transformado e N representa a altura do bloco.

O *ValorDC* será representado pelas Equações (2), (3) e (4). *refldx* e *p* nas Equações (2), (3) e (4) representam o índice de referência e a amostra vizinha, respectivamente. Como comentado anteriormente, o cálculo do somatório para o *ValorDC* é muito similar em cada equação apresentada, o que diferencia cada

equação de *ValorDC* serão os tamanhos de bloco. A Equação (2) será utilizada para tamanhos de bloco em que N e M possuem o mesmo tamanho. A Equação (3) é utilizada para tamanhos de bloco em que M é maior que N . A Equação (4) é utilizada para tamanhos de bloco em que M é menor que N .

$$amostrapredita[x][y] = ValorDC \quad (1)$$

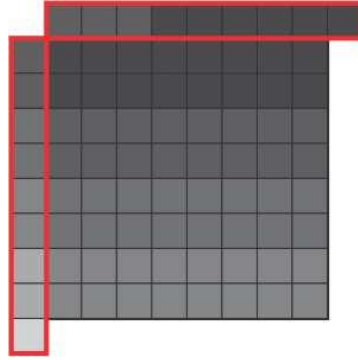
$$ValorDC = \left(\sum_{x'=0}^{M-1} p[x'][-1 - refIdx] + \sum_{y'=0}^{N-1} p[-1 - refIdx][y'] + M \right) > \\ > (Log_2(M) + 1) \quad (2)$$

$$ValorDC = \left(\sum_{x'=0}^{M-1} p[x'][-1 - refIdx] + (M \gg 1) \right) \gg Log_2(M) \quad (3)$$

$$ValorDC = \left(\sum_{y'=0}^{N-1} p[-1 - refIdx][y'] + (N \gg 1) \right) \gg Log_2(N) \quad (4)$$

O modo Planar cria uma predição suave para um bloco, assumindo que os valores de amostra variam pouco ao longo das direções horizontal e vertical. Para cada amostra no bloco a ser predito, o valor é calculado como uma combinação ponderada dos valores das amostras vizinhas à esquerda e acima do bloco. Isso resulta em uma transição suave entre os valores de amostras, ideal para áreas da imagem onde as variações de brilho são graduais. Como mostra a Figura 6, as amostras vizinhas mais próximas à amostra a ser predita possuem um peso maior comparado com amostras vizinhas mais longe da amostra a ser predita, gerando uma alteração mais suave entre uma amostra e outra. A parte destacada em vermelho na Figura 6 representa as amostras vizinhas usadas como referência na predição do bloco atual. No exemplo da Figura 6, é possível perceber que as amostras na parte de cima do bloco predito possuem um valor mais próximo às amostras vizinhas de cima, enquanto que as amostras na parte de baixo do bloco predito possuem um valor mais próximo às amostras vizinhas da esquerda.

Figura 6 - Processo do modo Planar da predição intra quadro



Fonte: Elaborada pelo autor (2024).

O cálculo realizado pelo modo Planar, pode ser visto na Equação (5). A Equação (5) mostra o cálculo de predição de uma amostra do bloco na posição $[x][y]$. Na Equação (5), $x = 0..M$ e $y = 0..N$, onde M representa a largura do bloco transformado e N representada a altura do bloco transformado. Como apresentado na Equação (5), o cálculo é dividido em $predV$ e $predH$, que representam as predições vertical e horizontal, respectivamente. O cálculo de $predV$ é apresentado na Equação (6), onde o deslocamento da predição tem como base M . O cálculo de $predH$ é apresentado na Equação (7), onde o deslocamento da predição tem como base N . Para ambas Equações (6) e (7) p representa a amostra vizinha utilizada na predição.

$$amostrapredita[x][y] = (predV[x][y] + predH[x][y] + M * N) \gg (Log_2(M) + Log_2(N) + 1) \quad (5)$$

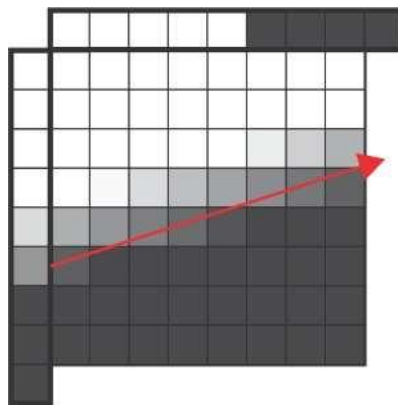
$$predV[x][y] = ((N - 1 - y) * p[x][-1] + (y + 1) * p[-1][N]) \ll Log_2(M) \quad (6)$$

$$predH[x][y] = ((M - 1 - x) * p[-1][y] + (x + 1) * p[M][-1]) \ll Log_2(N) \quad (7)$$

Os modos angulares (modos 2 a 66) utilizam seu respectivo ângulo como base para gerar a predição do bloco, deslocando e replicando os valores das amostras vizinhas para o bloco predito. Os modos horizontais (modos 2 a 34) deslocam apenas os valores das amostras vizinhas à esquerda do bloco, enquanto os modos verticais

(35 a 66) deslocam apenas os valores das amostras acima do bloco. A Figura 7 mostra um exemplo do processo de predição realizado pelos modos angulares. O exemplo em questão é de um modo angular horizontal, o modo 14 mais especificamente. Como pode ser visto na Figura 7, o bloco é predito através de deslocamento e replicação dos valores das amostras à esquerda do bloco atual, seguindo como base o ângulo do modo angular 14, representado na Figura 7 pela reta em vermelho. É importante ressaltar que, por se tratar de um modo angular horizontal, as amostras vizinhas acima do bloco não influenciam na predição do bloco, como explicado anteriormente. Quando um modo angular vertical é utilizado, o processo se inverte e apenas as amostras vizinhas acima do bloco são utilizadas para predição do bloco.

Figura 7 - Processo dos modos angulares da predição intra quadro. Exemplo do modo angular 14



Fonte: Elaborada pelo autor (2024).

O cálculo realizado pela predição intra quadro dos modos angulares pode ser visto na Equação (8). A Equação (8) mostra o cálculo de predição de uma amostra do bloco na posição $[x][y]$, onde o *Fator* é o fator de multiplicação baseado em um fator angular presente em cada modo angular. Esse fator angular possui um valor constante para cada modo angular. O valor de cada fator angular é apresentado na Tabela 1, onde o intervalo de valores possíveis para cada modo angular varia de -32 a 31. Além do fator de multiplicação, a Equação (8) utiliza *Idx* e *ref* que são o índice de multiplicação e o vetor de referência, respectivamente.

$$amostraPredita[x][y] = ((32 - Fator) * ref[x + Idx + 1] + Fator * ref[x + Idx + 2] + 16) >> 5 \quad (8)$$

Para que seja possível entender a Equação (8), é necessário também entender como são gerados *ref*, *Idx* e *Fator*. O cálculo de *ref* está representado pela Equação (9) e Equação (10). A Equação (9) mostra como o cálculo é feito para o modo angular 34 ou modos angulares maiores que 34 (modos 34 a 66), onde *p* são amostras vizinhas ao bloco atual, *x* é a posição do vetor de referência e *M* é a largura do bloco. Como pode ser visto na Equação (9), o -1 presente tanto em *x* quanto em *y* de *p[x][y]* é justamente para garantir que o vetor de referência seja preenchido com amostras vizinhas posicionadas logo acima do bloco atual. Na Equação (9), $x = 0..M + refIdx + 1$. A Equação (10) funciona de modo muito similar à Equação (9), com a diferença que os índices de *p* agora são invertidos para garantir que os modos angulares menores que o modo angular 34 (modos 2 a 33) utilizem amostras vizinhas à esquerda do bloco atual. Na Equação (10), $x = 0..N + refIdx + 1$.

$$ref[x] = p[-1 - refIdx + x][-1 - refIdx] \quad (9)$$

$$ref[x] = p[-1 - refIdx][-1 - refIdx + x] \quad (10)$$

O cálculo do *Idx* está representado na Equação (11) e na Equação (12). Para realizar o cálculo de *Idx* é necessário utilizar como base o fator angular baseado no modo angular como pode ser visto na Tabela 1. A Equação (11) serve para o modo angular 34 e os modos angulares maiores que o modo angular 34, onde o cálculo utiliza o valor de coluna *y*. Já a Equação (12) serve para modos angulares menores que o modo angular 34, utilizando o valor de linha *x* como base. Como o fator angular se baseia em valores em um intervalo de -32 a 31, parte da equação é dividida por 32 (com deslocamento à direita de cinco bits).

$$Idx = (((y + 1 + refIdx) * FA) >> 5) + refIdx \quad (11)$$

$$Idx = (((x + 1 + refIdx) * FA) >> 5) + refIdx \quad (12)$$

Por fim, o cálculo do fator de multiplicação é representado pela Equação (13) e pela Equação (14). Assim como o índice, o *Fator* se baseia no fator angular apresentado na Tabela 1. A diferença entre a Equação (13) e a Equação (14) também é a mesma, onde a Equação (13) é utilizada para os modos angulares 34 ou acima e utiliza a coluna *y* como referência, enquanto que a Equação (14), é usada para modos angulares menores que o modo 34 e utiliza a linha *x* como referência para o cálculo.

$$Fator = ((y + 1 + refldx) * FA) \& 31 \quad (13)$$

$$Fator = ((x + 1 + refldx) * FA) \& 31 \quad (14)$$

Tabela 1 - Fator angular dos modos angulares da predição intra quadro do VVC, onde MA representa modo angular e FA representa fator angular

MA	2	3	4	5	6	7	8	9	10	11	12
FA	32	29	26	23	20	18	16	14	12	10	8
MA	13	14	15	16	17	18	19	20	21	22	23
FA	6	4	3	2	1	0	-1	-2	-3	-4	-6
MA	24	25	26	27	28	29	30	31	32	33	34
FA	-8	-10	-12	-14	-16	-18	-20	-23	-26	-29	-32
MA	35	36	37	38	39	40	41	42	43	44	45
FA	-29	-26	-23	-20	-18	-16	-14	-12	-10	-8	-6
MA	46	47	48	49	50	51	52	53	54	55	56
FA	-4	-3	-2	-1	0	1	2	3	4	6	8
MA	57	58	59	60	61	62	63	64	64	66	
FA	10	12	14	16	18	20	23	26	29	32	

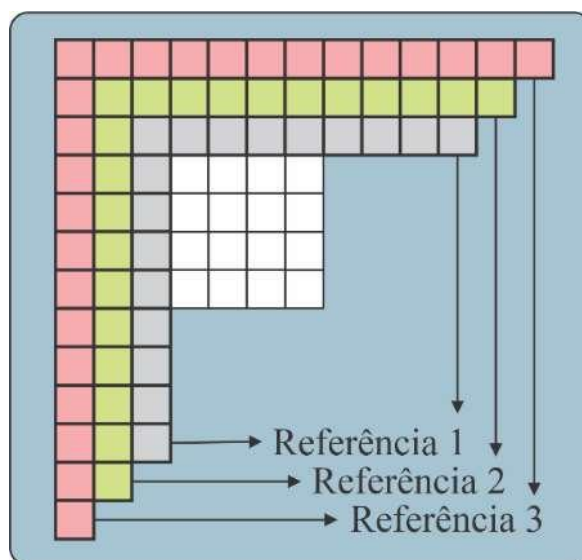
Fonte: Elaborada pelo autor (2024).

A seguir, as demais ferramentas da predição intra quadro do VVC serão apresentadas de forma mais superficial, uma vez que não são foco deste trabalho.

A ferramenta MRL, como pode ser visto na Figura 8, adiciona duas novas linhas de referência no processo de predição de bloco (linhas de referência 2 e 3). Essas duas novas linhas são incluídas com o objetivo de melhorar a eficiência de codificação

da predição intra quadro, uma vez que mesmo linhas de referência adjacentes podem diferir significativamente por causa de descontinuidades, levando à erros significativos de predição.

Figura 8 - Ferramenta de múltiplas linhas de referência da predição intra quadro do padrão de codificação VVC



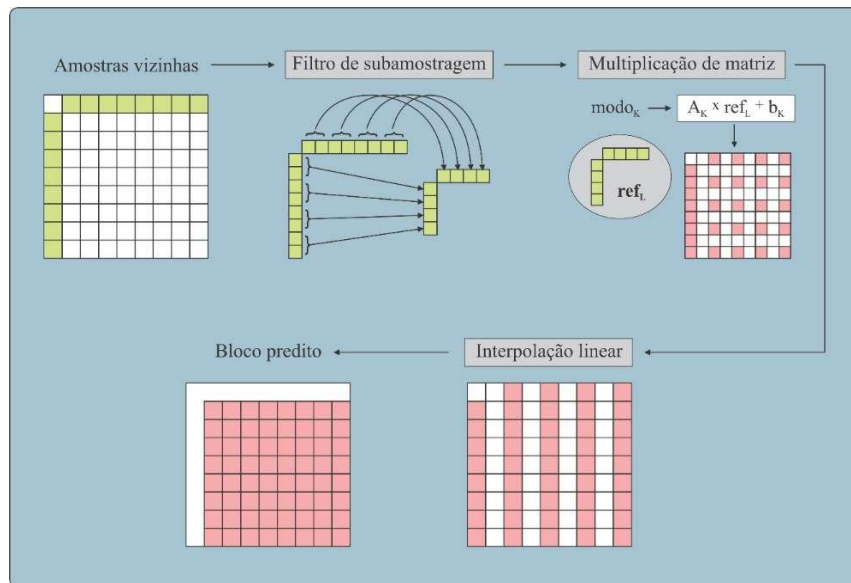
Fonte: Elaborada pelo autor (2024).

MRL é uma ferramenta que avalia cada combinação de modo de predição usando RMD e atualizando a lista RD (que já contém os melhores modos angulares inclusos). Porém, utilizar essa ferramenta para avaliar todos os modos de predição com a adição de duas linhas de referência teria um elevado custo computacional. Como forma de mitigar esse custo, a ferramenta MRL avalia somente MPMs com as duas linhas extras de referência (PFAFF et al., 2021). É relevante comentar que o modo planar não é considerado na ferramenta MRL, pois a adição da ferramenta ao modo não provê nenhum ganho de codificação adicional para a predição intra quadro do codificador VVC (PFAFF et al., 2021).

A ferramenta MIP representa um novo conceito em predição intra quadro com utilização de métodos baseado em aprendizado profundo, sendo uma alternativa aos modos de predição convencionais da predição intra quadro (PFAFF et al., 2021). Como pode ser visto na Figura 9, essa ferramenta realiza a predição de bloco através de multiplicação de matriz e interpolação de amostras.

O processo começa com a entrada de amostras vizinhas para a predição. Essas amostras vizinhas passam por um filtro de subamostragem para realização da multiplicação em matriz, como mostra a Figura 9. Após, é adicionado um desvio (b_k) e uma interpolação linear (horizontal e vertical $-A_k$) para obtenção do bloco predito (SCHÄFER et al., 2019).

Figura 9 - Ferramenta de predição intra quadro baseada em matriz do padrão de codificação VVC



Fonte: Elaborada pelo autor (2024).

MIP possui conjuntos de matrizes que foram definidos para cada tamanho de bloco para essa ferramenta através de treinamento de redes neurais *offline* e cada matriz representa um modo de predição. Assim como a MRL, a MIP também é avaliada usando RMD e atualiza a lista RD com a menor distorção entre a AIP, a MRL e a MIP (CHEN; YE; KIM, 2020). O diferencial da ferramenta MIP é melhorar a eficiência de codificação em predições não lineares, que seriam impossíveis com os modos angulares convencionais.

A ferramenta ISP explora as correlações entre amostras de bloco para melhorar a predição intra quadro (DE-LUXÁN-HERNÁNDEZ et al., 2019). Essa ferramenta divide o bloco atual em subpartições horizontais ou verticais que serão codificadas sequencialmente usando o mesmo modo de predição intra quadro. A ordem de processamento das subpartições é realizada da subpartição mais acima para a subpartição mais abaixo (para as divisões horizontais) e da subpartição mais à

esquerda para a subpartição mais à direita (para divisões verticais). Diferente do modo convencional, em que as amostras de referência usadas para a predição do bloco são apenas as amostras à esquerda e acima do bloco a ser predito, a ISP utiliza as amostras reconstruídas das subpartições já codificadas como amostras de referência para a codificação da próxima subpartição, aumentando o número de amostras de referência (DE-LUXÁN-HERNÁNDEZ et al., 2019).

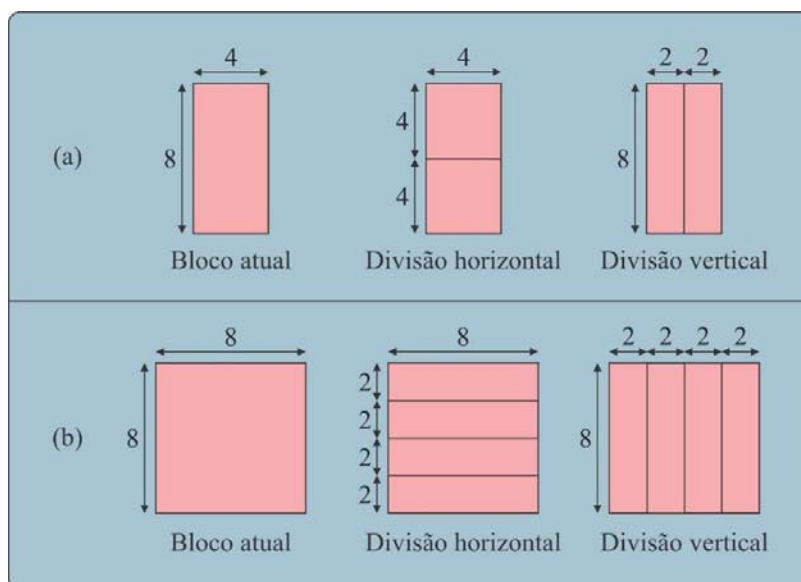
O tamanho de bloco 4x4 não é utilizado no processo da ferramenta ISP. Os tamanhos de bloco 8x4 e 4x8 são divididos em duas subpartições (verticais ou horizontais), como na

Figura 10.(a), que representa um exemplo de divisão para o tamanho de bloco 8x4. Os outros tamanhos de bloco disponíveis são divididos em quatro subpartições ao invés de apenas duas (DE-LUXÁN-HERNÁNDEZ et al., 2019). A

Figura 10.(b) mostra um exemplo de bloco de tamanho 8x8 sendo dividido em quatro subpartições de 2x8 na divisão horizontal e quatro subpartições de 8x2 na divisão vertical. O exemplo da

Figura 10.(b) representa não só o tamanho de bloco 8x8, como também todos os outros tamanhos de bloco que passam por quatro subpartições.

Figura 10 - Ferramenta de subpartição intra quadro do padrão de codificação VVC



Fonte: Elaborada pelo autor (2024).

Legenda: (a) Exemplo de bloco 8x4 dividido em 4x4 horizontal e 8x2 vertical. (b) Exemplo de bloco 8x8 dividido em 2x8 horizontal e 8x2 vertical.

Como a ferramenta ISP utiliza amostras reais reconstruídas como referência para as próximas predições de subpartição, essa ferramenta não pode utilizar o processo RMD, que somente pode ser obtido através de processo RDO. Para isso, o VTM utiliza estratégias para utilizar os modos de predição mais promissores. Primeiramente, o processo RDO é realizado com a lista RD contendo os melhores custos entre os modos AIP (como MRL e MIP não são considerados na ferramenta ISP, seus custos são descartados). Com esses custos, é gerada a lista de candidatos mais promissores para a ferramenta ISP. A lista é gerada alternando entre as divisões horizontal e vertical seguindo a ordem: planar, modos angulares ordenados por custo RD, DC e os melhores custos do AIP descartados da lista RD após o processamento de MRL e MIP. A lista ISP processa até 16 modos de predição, divididos em oito modos para a divisão horizontal e oito modos para a divisão vertical (DE-LUXÁN-HERNÁNDEZ et al., 2019).

3 Trabalhos Relacionados

Na seção anterior, foi discutido o codificador VVC, com ênfase na predição intra quadro, devido ao fato desse trabalho ser focado nessa ferramenta do codificador VVC. Desde o momento da criação do padrão de codificação VVC até os dias atuais, diversos trabalhos voltados à predição intra quadro desse padrão foram apresentados na literatura através de conferências e revistas da área de codificação de vídeos digitais, propondo múltiplas alterações e melhorias para essa ferramenta.

O trabalho de (ZHANG et al., 2020) por exemplo, propõe um método de decisão rápida de tamanhos de CU para diminuir o custo computacional da partição de blocos para a predição intra quadro do VVC. Segundo (ZHANG et al., 2020), o método proposto se divide em: um algoritmo de decisão rápida de partição de CU baseado no valor de textura e um algoritmo de decisão de partição de CU baseado na direção da textura.

As características presentes na textura são usadas como base para prever a partição de CU, descartando partições não candidatas durante o processo de codificação do VVC. O algoritmo proposto por (ZHANG et al., 2020) foi implementado no VTM versão 7.0. Os experimentos gerados desse trabalho mostram uma redução de custo computacional de aproximadamente 48,58%. Em eficiência de codificação, o trabalho de (ZHANG et al., 2020) apresentou um aumento da taxa *Bjontegaard Delta Rate* (BD-Rate) de cerca de 0,91%.

O trabalho de (LIU et al., 2023) apresenta uma solução na parte de decisão de modos para a predição intra quadro do VVC utilizando inteligência artificial. (LIU et al., 2023) propõe uma abordagem de decisão rápida de modos intra quadro baseada em aprendizagem multitarefa profunda por meio de poda adaptativa dos modos mais redundantes. Primeiro é criada uma base de dados de todos os modos intra quadro em larga escala. Após, uma rede de decisão de modos intra quadro multitarefa é proposta para prever precisamente os modos angulares mais prováveis e se os modos ISP e MIP devem ser pulados. A seguir, (LIU et al., 2023) utiliza um fluxo rápido de codificação intra quadro, envolvendo aceleração de decisão RMD e poda de modos da lista de candidatos. Na saída do fluxo, a probabilidade da aprendizagem e a probabilidade estatística são sintetizadas juntas para melhorar a precisão da predição, garantindo que apenas os modos intra quadro desnecessários sejam pulados. O

trabalho proposto por (LIU et al., 2023) apresentou três resultados diferentes dependendo dos parâmetros utilizados, onde a redução do tempo de codificação da predição intra quadro atingiu os valores 16,72%, 30,12% e 40,48% do mais lento para o mais rápido. Já a parte de eficiência de codificação, na mesma ordem listada acima os resultados de BD-Rate foram 0,83%, 1,56% e 2,66%, respectivamente.

Outro trabalho, como o de (LIN et al., 2022) propõe uma predição intra quadro de alta ordem. Segundo (LIN et al., 2022) essa alta ordem é modelada através de uma função de extrapolação quadrática, com o objetivo de gerar padrões de alta fidelidade com estruturas não lineares, algo que não é possível com os preditores lineares do codificador VVC. Para ser compatível com o atual mecanismo de predição intra quadro, a função quadrática pode ser formulada com o uso de dois modos angulares. Para reduzir a complexidade de codificação, (LIN et al., 2022) ainda propõe uma estratégia de poda de modos afim de encontrar os pares de modos mais apropriados para a função. Os resultados apresentados por (LIN et al., 2022) mostram uma melhoria na eficiência de codificação de até 0,6% (redução de BD-Rate) com aumento moderado no custo computacional do codificador.

Diversas pesquisas foram feitas com foco na predição intra quadro do padrão VVC, onde algumas foram citadas logo acima, com diversas mudanças em partes da predição intra quadro como a partição de blocos, decisões de modo de predição e criação de novas ferramentas. Mas praticamente não existem trabalhos com foco em implementações de hardware para a ferramenta modos angulares da predição intra quadro do padrão de codificação VVC. Devido à falta desses trabalhos, a comparação entre o trabalho proposto e a literatura torna-se uma tarefa árdua. Por causa dessa falta de trabalhos, algumas propostas de hardware para os modos angulares do HEVC serão referenciadas neste trabalho afim de realizar uma comparação de resultados. É importante salientar que a comparação entre dois padrões de codificação diferentes é extremamente difícil pela diferença de eficiência de codificação apresentada por cada padrão.

O trabalho de (HUANG et al., 2016) foca em algoritmos rápidos para redução de custo computacional e dependência de dados para os modos angulares da predição intra quadro do HEVC. Dentre as melhorias apresentadas por (HUANG et al., 2016) para eliminar a dependência de dados estão: rápidos algoritmos de RMD, modo de decisão RDO, adaptação de contexto paralelizado e decisão de CU/PU livre

de croma. Além disso, uma arquitetura VLSI paralela para os algoritmos propostos é apresentada por (HUANG et al., 2016) para a decisão de modos intra quadro do HEVC. A heurística para modos intra quadro proposta por (HUANG et al., 2016) atingiu uma redução de 41,6% de custo computacional ao custo de uma aumento no BD-Rate de 4,3% em média. A arquitetura projetada para a tecnologia CMOS 55 nm necessita de uma área de 1.571,7 k portas NAND2 equivalentes. Os resultados de síntese mostram que a arquitetura é capaz de alcançar o processamento de vídeos 1080p@60qps com uma frequência de 294 MHz.

O trabalho de (ZHANG; LU, 2019) propõe quatro algoritmos de adaptação com foco em redução de dependência de dados e custo computacional para os modos angulares da predição intra quadro do HEVC. Dentre as técnicas propostas para esses algoritmos estão a inclusão de um modo de pré-seleção para unidades de predição (do inglês *prediction unit* - PU) em croma e luma, modificação dos modos de decisão de CUs e simplificação da estimativa da taxa de codificação aritmética binária adaptativa baseada em contexto (do inglês *context-based adaptive binary arithmetic coding* - CABAC) baseada em tabela. Utilizando o software de referência *HEVC Test Model* (HM) (SUEHRING, 2018), os algoritmos propostos por (ZHANG; LU, 2019) demonstraram uma redução no custo computacional de cerca de 27%. Em termos de eficiência de codificação, o trabalho de (ZHANG; LU, 2019) apresentou um aumento no BD-Rate de 4,39%. Por fim, (ZHANG; LU, 2019) ainda propõe uma arquitetura de hardware totalmente paralela para os algoritmos propostos. Para alcançar alta precisão de performance foram implementadas múltiplas instâncias de estimação de taxa CABAC, onde 16 elementos de sintaxe são classificados em cinco grupos independentes para processamento paralelo. O trabalho proposto suporta todos os tamanhos de bloco e todos os modos angulares. A arquitetura é baseada na tecnologia TSMC 90 nm e é capaz de processar vídeos em resolução 2160@30qps. A arquitetura de (ZHANG; LU, 2019) apresenta resultados em implementação ASIC com 2.288 k portas NAND2 equivalentes de área e 236 mW de consumo energético para resolução 2160@30qps.

Diferente dos trabalhos apresentados anteriormente, o trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) é baseado na implementação original do VVC, assim como uma das arquiteturas propostas neste trabalho. Porém, devido a uma diferença de tecnologias utilizadas em cada trabalho, a comparação de resultados na parte de

hardware torna-se difícil. (AZGIN; KALALI; HAMZAOGLU, 2019) apresenta uma arquitetura dedicada para os modos angulares da predição intra do VVC. O hardware suporta os tamanhos de bloco 4x4, 8x8, 16x16 e 32x32. A arquitetura focada na tecnologia FPGA Virtex 7 implementa as quatro multiplicações constantes presentes na equação de um modo angular da predição intra quadro em dois blocos de processamento de sinal digital (do inglês *Digital Signal Processing* - DSP) (“Digital Signal Processing (DSP) Block”, [s.d.]) e dois somadores. Esse trabalho proposto por (AZGIN; KALALI; HAMZAOGLU, 2019) requer uma área de 46.382 tabelas de consulta (do inglês *Lookup Table* - LUT). No pior caso, a arquitetura processa vídeos em resolução 1080p@30qps rodando a uma frequência de 119 MHz.

Como apresentado acima, trabalhos com implementação de hardware para a ferramenta dos modos angulares da predição intra quadro do codificador VVC são praticamente inexistentes na atual literatura. Considerando esse fato, é possível entender a importância do trabalho realizado nessa dissertação, onde diferentes implementações em hardware para esta ferramenta são desenvolvidas. O principal objetivo do desenvolvimento dessas arquiteturas de hardware é analisar o impacto, os pontos positivos e os pontos negativos de cada arquitetura em comparação com outras, em um cenário em que arquiteturas de hardware para essa ferramenta ainda são escassas.

4 Heurísticas Desenvolvidas

Essa seção apresenta os aspectos do desenvolvimento das heurísticas utilizadas para a realização desse trabalho. Para uma melhor organização, essa seção foi dividida em: configuração experimental, resultados preliminares, heurística RFCC e heurística RBCC, onde serão apresentadas as estratégias utilizadas a nível de software para redução do custo computacional da ferramenta de modos angulares do VVC.

Para que fosse possível o desenvolvimento de heurísticas voltadas aos modos angulares da predição intra quadro do VVC, algumas avaliações preliminares foram feitas a fim de entender o comportamento tanto dos modos angulares, quanto dos tamanhos de blocos que são utilizados nessa ferramenta. Para isso, essa seção apresentará as condições de realização dos testes (configuração experimental), resultados preliminares sobre a ferramenta modos angulares do VVC, além de discorrer sobre cada heurística desenvolvida para esse trabalho.

4.1 Configuração Experimental

Os experimentos realizados durante esse trabalho consideram as definições impostas pelo documento Condições Comuns de Teste, do inglês *Common Test Conditions* (CTC) (BOSSSEN et al., 2019). Para realização dos testes, foi utilizado o VTM na versão 23.0 (BROSS et al., 2020), com a configuração *All-Intra* (AI). Essa configuração do codificador permite que apenas a predição intra quadro seja usada para codificar os quadros. Assim, cada quadro é codificado de forma independente dos demais. Embora esse seja um cenário não realista para aplicações reais, a configuração *All-Intra* foi desenvolvida exatamente para isolar a predição intra quadro e, assim, avaliar melhor as propostas de melhorias para essa ferramenta (BOSSSEN et al., 2019). Como esse trabalho é totalmente focado na predição intra quadro, usar essa configuração é a opção natural.

Também foram considerados quatro parâmetros de quantização, do inglês *Quantization Parameters* (QPs), conforme definido nas CTCs. Os QPs escolhidos foram: 22, 27, 32 e 37 (BOSSSEN et al., 2019). Ao todo, 26 sequências de vídeos, também definidas nas CTCs, foram utilizadas para os testes, com dez segundos de

vídeo cada uma. Essas sequências são divididas em seis classes devido as suas diferentes resoluções. Essas são:

- Classe A, de resolução 3840x2160, com as seguintes sequências: Tango2, FoodMarket4, CampFire, CatRobot, DaylightRoad2 e ParkRunning3;
- Classe B, de resolução 1920x1080, com as seguintes sequências: MarketPlace, RitualDance, Cactus, BasketballDrive e BQTerrace;
- Classe C, de resolução 832x480, com as seguintes sequências: RaceHorses, BQMall, PartyScene e BasketballDrill;
- Classe D, de resolução 416x240, com as seguintes sequências: RaceHorses, BQSquare, BlowingBubbles e BasketballPass;
- Classe E, de resolução 1280x720, com as seguintes sequências: FourPeople, Johnny e KristenAndSara;
- Classe F, de resoluções que variam de 832x480 até 1920x1080, com as seguintes sequências: ArenaOfValor, BasketballDrillText, SlideEditing e SlideShow;

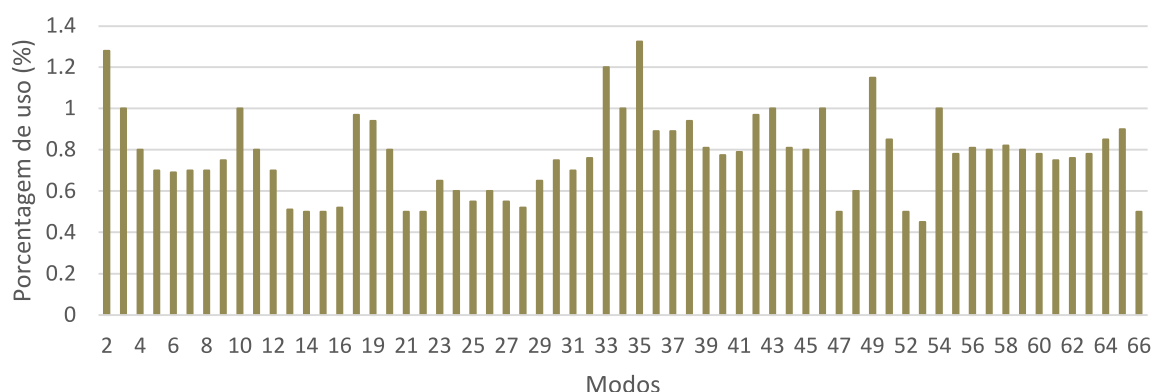
Para realização dos testes foi utilizado um servidor com dois processadores da série Xeon Gold 6430, com frequência de 2.10 GHz. Cada processador consiste em 32 núcleos físicos e 64 threads. O servidor é equipado com quatro pentes de memória RAM AMD DIMM de 32 GB, totalizando 128 GB de memória RAM. As memórias possuem uma frequência nominal de 4.400 MHz. O sistema operacional presente no servidor é o Linux Ubuntu na versão 20.04.6.

4.2 Avaliações Preliminares

Antes de definir qualquer heurística para os modos angulares do VVC, foi realizado um conjunto de experimentos para entender melhor o impacto que cada modo angular e cada tamanho de bloco possui dentro da ferramenta do codificador. A Figura 11 apresenta a porcentagem de uso de cada modo angular em relação às amostras de vídeo. Em outras palavras, qual percentual de amostras do vídeo foi codificado com cada modo angular. Como pode ser notado na Figura 11, os modos

Planar, DC, 18 e 50 não são apresentados. Isso se deve pelo fato destes serem os modos mais utilizados, o que tornaria difícil uma análise mais profunda dos modos menos utilizados. O modo Planar (ou modo 0) possui uma porcentagem de uso de 38%, sendo o modo mais utilizado. O segundo modo mais utilizado é o DC (ou modo 1) com 5% de uso. Os modos angulares 18 (ou modo horizontal) e 50 (ou modo vertical) são usados em 3,48% e 4,55% das amostras, respectivamente. Ao todo, esses quatro modos são os que mais se destacam dentre todos os modos disponíveis. Alguns modos, como os modos 2, 3, 10, 33, 34, 35, 43, 46, 49 e 54 possuem uma porcentagem uso de mais de 1%. O resto dos modos possuem uma taxa de uso que varia entre 0,5% e 1%.

Figura 11 - Porcentagem de uso dos modos angulares no VTM



Fonte: Elaborada pelo autor (2024).

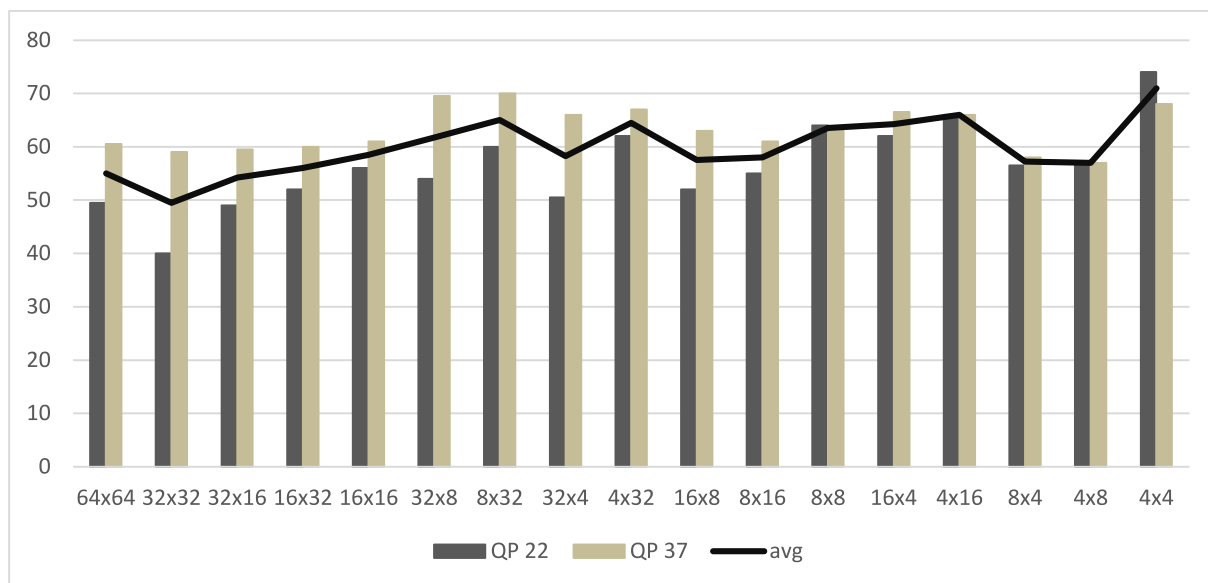
Além da análise realizada em torno dos modos angulares, também foi avaliada a importância de cada tamanho de bloco disponível para essa ferramenta do VVC. Essa análise foi realizada no trabalho de (SALDANHA et al., 2021) e, portanto, não foi repetida neste trabalho. Segundo sua análise, os tamanhos de bloco considerados quadrados (tamanhos de bloco com o mesmo número de linhas e colunas - 4x4, 8x8, 16x16, 32x32 e 64x64) são usados em cerca de 40% das amostras dos vídeos, enquanto que os outros 12 tamanhos de bloco disponíveis (4x8, 8x4, 4x16, 16x4, 8x16, 16x8, 4x32, 32x4, 8x32, 32x8, 16x32 e 32x16) são usados em cerca de 60% das amostras (SALDANHA et al., 2021). Essa análise demonstra que mesmo com a inclusão de diversos tamanhos de blocos novos em comparação ao padrão

antecessor (HEVC), os tamanhos de bloco quadrados ainda são bastante utilizados por essa ferramenta no VVC.

A Figura 12 mostra a porcentagem de uso dos modos angulares da predição intra quadro do padrão VVC para cada tamanho de bloco nos parâmetros de quantização 22 e 37 (mínimo e máximo definido pelas CTCs). A Figura 12 representa apenas a porcentagem relativa à escolha dos respectivos modos angulares na predição intra quadro para o tamanho de bloco respectivo. Ou seja, se um tamanho de bloco determinado descrito na Figura 12 tem uma porcentagem de uso de 60% para os modos angulares, significa que os outros 40% são para o resto das ferramentas de codificação disponíveis da predição intra quadro, como por exemplo, MRL, MIP ou ISP. Como a Figura 12 possui a análise de dois QPs diferentes, foi adicionado uma linha em preto à figura para representar a média de uso entre esses dois QPs.

Com a Figura 12, é possível perceber que, na média, os tamanhos de bloco médios ou menores tendem a ser mais codificados com modos angulares do que tamanhos de bloco maiores. O tamanho de bloco 4x4 é mais utilizado em média (cerca de 70% das vezes que esse bloco é utilizado na predição intra quadro), segundo a Figura 12. O tamanho de bloco menos utilizado em média pelos modos angulares é o 32x32 (cerca de 50% das vezes que esse bloco é utilizado na predição intra quadro). Quando considerado apenas o QP 22, um padrão similar ao anterior é observado, na média, com o tamanho 4x4 sendo utilizado cerca de 75% das vezes, enquanto o tamanho 32x32 é utilizado apenas 40% das vezes. Já quando considerado apenas o QP 37, é possível perceber uma diferença menor, com foco nos tamanhos de bloco médios. Os destaques positivos ficam com os tamanhos 32x8 e 8x32 (cerca de 70% das vezes), e os destaques negativos ficam para os tamanhos 8x4 e 4x8 (cerca de 55% das vezes).

Figura 12 - Porcentagem de uso dos tamanhos de bloco para os modos angulares utilizando os QPs 22 e 37 no VTM



Fonte: Elaborada pelo autor (2024).

4.3 Heurística de Redução Forte de Custo Computacional

A heurística de Redução Forte de Custo Computacional (RFCC) foi a primeira desenvolvida e tem como objetivo a criação de uma solução que possibilite uma redução expressiva no custo computacional dos modos angulares e que possibilite o projeto de uma arquitetura mais simples. Para isso, os modos angulares e tamanhos de bloco menos utilizados foram eliminados a fim de reduzir significativamente o custo computacional da ferramenta do VVC.

Entre todos os modos angulares disponíveis, somente os modos com porcentagem de uso acima de 1% foram mantidos, considerando os resultados apresentados na Figura 11. Os 14 modos angulares que se encaixam nessa condição são os modos: Planar, DC, 2, 3, 10, 18, 33, 34, 35, 43, 46, 49, 50 e 54. Vale ressaltar que os modos Planar, DC, 18 e 50 possuem mais de 1% de uso, porém não estão na figura para melhorar a visão dos modos angulares restantes.

Uma análise mais detalhada dos modos selecionados indica que em alguns pontos existem buracos consideráveis entre um modo angular e outro, com muitos ângulos contínuos sendo descartados. Para reduzir essa diferença, os modos 7, 23, 26 e 30 foram adicionados à lista de modos selecionados, totalizando 18 modos

angulares. Comparando esse valor com a quantidade de modos disponíveis pelo VVC, ou seja, 67 modos, é possível notar uma redução de aproximadamente 73,13% no número de modos avaliados.

Baseado nas análises anteriores, a heurística RFCC também propõe uma redução de 17 tamanhos de bloco para apenas cinco tamanhos de bloco. Nesse caso, apenas os tamanhos de bloco quadrados foram selecionados (64x64, 32x32, 16x16, 8x8 e 4x4). A redução de tamanhos de bloco é de cerca de 70,59% do total de blocos avaliados. Entre modos angulares e tamanhos de bloco, a heurística RFCC apresenta uma redução total de aproximadamente 92,1% nas avaliações necessárias. Como cada modo angular é processado para todos os tamanhos de bloco, o cálculo de opções de codificação é $OC = M * T$, onde OC representa as opções de codificação, M representa os modos angulares e T representa os tamanhos de bloco. Utilizando esse cálculo como base, o VTM possui 1139 opções de codificação disponíveis, enquanto a heurística RFCC possui apenas 90 opções de codificação disponíveis. Ou seja, a heurística RFCC utiliza apenas 7,9% das opções disponíveis originalmente pelo VTM. Naturalmente é esperado que um corte tão intenso na ferramenta gere um impacto significativo na eficiência de codificação.

Os experimentos realizados para essa heurística foram baseados no documento CTC, utilizando as especificações citadas anteriormente. Dois resultados foram considerados para avaliar o impacto da heurística RFCC. O primeiro resultado é a métrica BD-Rate (BJONTEGAARD, 2008), onde o valor de BD-Rate, apresentado em forma de porcentagem, representa o aumento na taxa de bits necessário para manter a qualidade objetiva da imagem a partir de alguma modificação no codificador VTM. Esse valor pode ser considerado como perda na eficiência de codificação, caso o valor de BD-Rate seja positivo, ou ganho na eficiência de codificação, caso o valor de BD-Rate seja negativo (BJONTEGAARD, 2008). O segundo resultado é a redução de tempo de codificação (RTC), que representa o aumento ou a redução no custo computacional do codificador completo, quando são feitas alterações na ferramenta de predição intra quadro. Quanto maior o RTC, maior é a redução de custo computacional do codificador. O RTC é obtido através de uma comparação entre o tempo de codificação obtido pelo VTM sem qualquer alteração e o tempo de codificação obtido pelo VTM utilizando a heurística definida. Assim, o RTC considera

também os custos de outras ferramentas, como transformadas, codificação de entropia, filtros e mesmo outras ferramentas avaliadas na predição intra quadro.

Os resultados estão apresentados na Tabela 2. Para uma melhor visualização os dados estão organizados por classes de sequências de vídeo, ou seja, os resultados apresentados para cada classe representam a média de resultados das sequências de vídeos pertencentes a cada classe, para os quatro QPs avaliados. Cada sequência de vídeo foi codificada com os quatro QPs. Os resultados de BD-Rate e RTC de cada sequência de vídeo foram obtidos através de uma média dos resultados extraídos de cada QP.

Tabela 2 - Média de BD-Rate e RTC por classe de sequências de vídeo do CTC para heurística RFCC

Classe	Resolução	Quantidade de Sequências	BD-Rate	RTC
A	3840x2160	6	5,40%	60,34%
B	1920x1080	5	8,92%	73,34%
C	832x480	4	7,85%	50,75%
D	416x240	4	7,03%	57,38%
E	1280x720	3	10,93%	65,51%
F	832x480 até 1920x1080	4	11,60%	58,22%
Média			8,62%	60,92%

Fonte: Elaborada pelo autor (2024).

Como pode ser observado na Tabela 2, o menor BD-Rate foi registrado pela classe A, com 5,4%, enquanto o maior BD-Rate foi da classe F com 11,6%. A média de todas as classes é de 8,62% de BD-Rate. Já na questão de custo computacional, a heurística RFCC apresentou o pior RTC na classe C, com 50,75% de redução de tempo, enquanto a classe B registrou o melhor resultado, com 73,34% de redução de tempo. A média de RTC registrada por todas as classes foi de 60,92%. Aqui é importante destacar que mesmo com uma redução de opções de codificação (OC) de

92,1% na ferramenta de modos angulares da predição intra quadro, conforme discutido anteriormente, essa redução tem um impacto de 60,92% no tempo total do codificador, uma vez que, também como já comentado, o RTC considera o tempo do codificador completo, ou seja, com todas as demais ferramentas de codificação ativas. Quando considerados ambos os resultados em conjunto, a classe A, de vídeos 4K (3840x2160), possui os melhores resultados. Segundo a Tabela 2, esta classe possui a menor perda na eficiência de codificação (5,4% de BD-Rate) e a terceira maior redução de custo computacional (60,34% de RTC).

É importante ressaltar que, o valor médio de aumento de BD-Rate de 8,62% pode ser considerado um valor alto. Porém, essa heurística tem como foco uma expressiva redução de custo computacional da ferramenta. Apesar de possuir um valor elevado de BD-Rate, a heurística também possui um valor elevado de redução de custo computacional (média de 60,92% de redução), cumprindo com seu objetivo.

4.5 Heurística de Redução Balanceada de Custo Computacional

A heurística de Redução Balanceada de Custo Computacional (RBCC) tem como foco tanto a redução de custo computacional, quanto a máxima eficiência de codificação. Para que seja possível chegar em uma heurística que reduza significativamente o custo computacional mantendo ao máximo a qualidade de vídeo, mais algumas avaliações foram realizadas no VTM para entender o impacto da retirada de modos angulares e tamanhos de bloco na qualidade do vídeo.

Como pode ser visto na Tabela 3, seis avaliações foram realizadas no VTM, retirando modos angulares, para analisar o impacto dessas remoções na qualidade do vídeo. Nas avaliações realizadas (M1, M2, M4, M14, M27, M41), o M representa os modos angulares e o número que está à direita do M representa a quantidade de modos angulares retirados para a avaliação. É importante ressaltar que a ordem de modos angulares retirados em cada experimento condiz com a análise realizada na subseção de avaliações preliminares (Figura 11). Ou seja, no experimento M1 foi retirado o modo angular menos utilizado, no experimento M2 foram retirados os dois modos angulares menos utilizados e assim por diante.

Quando são analisados os resultados para M1, M2 e M4, é possível perceber um aumento muito pequeno no BD-Rate. Mesmo com o aumento na redução de

modos angulares entre os experimentos M1, M2 e M4, a mudança foi tão desprezível que não pôde ser representada com apenas duas casas decimais. Os melhores resultados foram vistos nas classes B e C, com um aumento de 0,02% no BD-Rate, e o pior resultado foi registrado na classe E, com 0,06% de aumento no BD-Rate. Em média, o aumento do BD-Rate desses três experimentos chegou à 0,04%, de acordo com a Tabela 3.

Tabela 3 - Análise de BD-Rate sobre redução de modos angulares para a predição intra quadro

Classe	Resolução	Quantidade de Sequências	BD-Rate (%)					
			M1	M2	M4	M14	M27	M41
A	3840x2160	6	0,04	0,04	0,04	0,13	0,25	0,48
B	1920x1080	5	0,02	0,02	0,02	0,09	0,27	0,44
C	832x480	4	0,02	0,02	0,02	0,09	0,26	0,48
D	416x240	4	0,04	0,04	0,04	0,12	0,35	0,53
E	1280x720	3	0,06	0,06	0,06	0,13	0,43	0,78
F	832x480 até 1920x1080	4	0,05	0,05	0,05	0,1	0,19	0,41
Média			0,04	0,04	0,04	0,11	0,29	0,52

Fonte: Elaborada pelo autor (2024).

No experimento M14 é realizada uma redução mais significativa nos modos angulares disponíveis. Com esses quatorze modos retirados, o M14 representa uma redução de cerca de 20,9% do total de modos disponíveis. Como pode ser visto na Tabela 3, a média de aumento no BD-Rate para o M14 é de 0,11%, onde as classes B e C obtiveram o melhor resultado (0,09% de aumento no BD-Rate), enquanto as classes A e E obtiveram os piores resultados (0,13% de aumento no BD-Rate).

No experimento M27, a redução de modos utilizados passa a ser de cerca de 40,3% no total de modos angulares disponíveis. A partir desse experimento, é possível perceber a criação de alguns espaços no intervalo de modos angulares, onde algumas partes do intervalo de ângulos descartam uma sequência de cinco ou mais modos

angulares. Como, por exemplo, os ângulos entre os modos 21 e 31 são retirados por estarem entre os menos utilizados. Quando é analisado o aumento de BD-Rate médio do experimento M27 (0,29% de aumento no BD-Rate) e esse aumento médio é comparado com o aumento médio de BD-Rate do experimento anterior (M14), é possível perceber um aumento significativo no M27. Essa diferença pode ser justificada pelo descarte de muitos modos angulares em sequência, como já discutido. A classe de vídeos com melhor performance no M27 foi a classe F com um aumento de 0,19% de BD-Rate, enquanto as piores performances foram detectadas nas classes D e E com 0,35% e 0,43% de aumento no BD-Rate, respectivamente.

O experimento M41 segue um padrão parecido com o que foi o experimento M27, agora com uma redução de cerca de 61,19% do total de modos angulares disponíveis para a predição intra quadro. Como a redução passou de 50% dos modos disponíveis para esse experimento, a ferramenta modos angulares passa a ter modos esparsos. Por exemplo, no intervalo entre os modos 11 e 32, agora apenas três modos são considerados para representar esses ângulos. O melhor desempenho do experimento M41 foi detectado na classe de sequências de vídeo F, com 0,41% de aumento no BD-Rate, conforme apresentado na Tabela 3. A classe B possui o segundo melhor desempenho, apenas 0,44% de aumento no BD-Rate. O pior desempenho desse experimento foi registrado pela classe E, com um aumento de 0,78% no BD-Rate. Uma diferença de 0,25% em relação ao segundo pior desempenho (0,53% de aumento de BD-Rate), registrado pela classe D. Segundo a Tabela 3, o aumento médio do experimento foi de 0,52%.

Assim como na análise de impacto sobre a remoção de alguns modos angulares, experimentos para análise de impacto sobre a remoção de alguns tamanhos de bloco também foram realizados. Ao total, seis testes foram realizados no VTM retirando tamanhos de bloco da predição intra quadro a fim de analisar o impacto dessas remoções na eficiência de codificação (representados na Tabela 4). Nos testes realizados (B1, B2, B3, B4, B7, B10), o B representa os tamanhos de bloco e o número que está à direita do B representa a quantidade de tamanhos de bloco retirados para a avaliação. É importante ressaltar que a ordem de tamanhos de bloco retirados em cada experimento condiz com a análise realizada acima na subseção de resultados preliminares (Figura 12). Ou seja, no experimento B1 foi retirado o tamanho de bloco menos utilizado, no experimento B2 foram retirados os dois tamanhos de

bloco menos utilizados e assim por diante. Também é importante ressaltar que a análise de tamanhos de bloco menos utilizados foi feita baseada no tamanho de uma CTU, que no caso da predição intra quadro é de 64x64. Ou seja, um bloco de tamanho 64x64 equivale a um uso em uma CTU e 256 blocos 4x4 também equivalem a um uso em uma CTU (considerando que são necessários 256 blocos 4x4 para preencher uma CTU 64x64).

Quando são comparados o impacto da retirada de um modo angular e o impacto da retirada de um tamanho de bloco, é possível perceber o quão significativo é o aumento no BD-Rate quando são removidos os tamanhos de bloco, prejudicando a eficiência de codificação. Como mostra a Tabela 4, o aumento médio do BD-Rate para o experimento B1 é de 0,56%. Esse valor pode não parecer tão alto em um primeiro momento, porém quando esse valor é comparado com os experimentos da Tabela 3, é possível perceber que a remoção de apenas um tamanho de bloco possui mais impacto na eficiência de codificação que a remoção de 60% dos modos angulares disponíveis para predição intra quadro (média de 0,52% de aumento do BD-Rate). O melhor desempenho no experimento B1 foi da classe C com 0,44% de BD-Rate, enquanto o pior desempenho foi registrado pela classe A com 0,64%. Seguido das classes B e E, com 0,61% de BD-Rate, segundo a Tabela 4.

Tabela 4 - Análise de BD-Rate sobre redução de tamanhos de bloco para a predição intra quadro

Classe	Resolução	Quantidade de Sequências	BD-Rate (%)					
			B1	B2	B3	B4	B7	B10
A	3840x2160	6	0,64	1,43	2,06	2,83	4,39	6,5
B	1920x1080	5	0,61	1,24	1,98	2,73	4,3	6,44
C	832x480	4	0,44	0,97	1,59	2,2	3,78	5,81
D	416x240	4	0,46	0,92	1,54	2,22	3,84	5,8
E	1280x720	3	0,61	1,01	1,7	2,31	3,96	5,96
F	832x480 até 1920x1080	4	0,6	1,37	1,99	2,61	4,2	6,28
Média			0,56	1,16	1,81	2,48	4,08	6,13

Fonte: Elaborada pelo autor (2024).

No experimento B2, onde dois tamanhos de bloco foram removidos, por se tratar de apenas 17 tamanhos de bloco disponíveis, a redução de tamanhos de blocos ultrapassa os 10% (uma redução de cerca de 11,77%). Como apresentado na Tabela 4, o BD-Rate médio para esse experimento foi de 1,16%, onde o pior desempenho foi registrado pela classe A, com 1,43% de BD-Rate, seguido das classes F e B, com 1,37% e 1,24% de BD-Rate, respectivamente. O melhor desempenho ficou com a classe D, com 0,92% de BD-Rate.

Como a remoção de cada tamanho de bloco representa uma redução de cerca de 5,88% nos tamanhos de bloco disponíveis, o experimento B3 possui uma redução de cerca de 17,65% dos tamanhos de bloco disponíveis. Essa redução tem um impacto de 1,81% no BD-Rate. Novamente, o pior desempenho registrado foi na classe A com 2,06% de BD-Rate. Seguido das classes F e B, agora muito próximas, com 1,99% e 1,98% de BD-Rate, respectivamente. O melhor desempenho desse experimento, segundo a Tabela 4, foi apresentado pelas classes C e D, com 1,59% e 1,54% de BD-Rate, respectivamente. É possível perceber, através da Tabela 4, que a redução dos tamanhos de bloco utilizados na predição intra quadro afeta mais sequências de maior resolução (4K e 1080p) do que de resoluções menores (480p e 240p).

A partir do experimento B4, com a remoção de quatro tamanhos de bloco, a redução ultrapassa a marca de 20% nos tamanhos de bloco disponíveis, mais precisamente uma redução de 23,53%. O melhor desempenho apresentado nesse experimento foi apresentado pelas classes C e D, com 2,2% e 2,22% de BD-Rate, respectivamente. O pior desempenho segue sendo da classe A (classe de sequências de vídeo 4K), com um BD-Rate de 2,83%. A média de todas as classes para B4, segundo a Tabela 4, foi de 2,48% de BD-Rate.

No experimento B7, há uma redução total de cerca de 41,18% nos tamanhos de bloco (mantendo 10 tamanhos de bloco dos 17 disponíveis). O BD-Rate para o experimento B7 foi de 4,08% na média. Um valor relativamente alto para os padrões atuais de pesquisa, onde muitos trabalhos são apresentados com aumentos de BD-Rate próximos à 1% ou 2%. O pior desempenho foi registrado pela classe A com 4,39%, seguido da classe B com 4,3% no BD-Rate. O melhor desempenho foi apresentado pela classe C com 3,78%, segundo a Tabela 4.

Por fim, o experimento B10 possui uma redução de tamanhos de bloco total de 58,82% aproximadamente. Como nesse experimento há uma redução maior que 50% nos tamanhos de bloco disponíveis para predição intra quadro, é esperado que o BD-Rate seja significativamente alto. Como pode ser visto na Tabela 4, esse valor é de 6,13% na média. Onde, as classes B e C, com os melhores desempenhos, obtiveram resultados muito próximos, 5,81% e 5,8% respectivamente. As classes com maior resolução seguem tendo os piores desempenhos, onde a classe A se destaca negativamente, com 6,5% de aumento no BD-Rate.

A heurística RBCC, como comentado no início dessa subseção, tem como objetivo reduzir o custo computacional da ferramenta de modos angulares da predição intra quadro do padrão de codificação VVC, porém, diferente do que foi apresentado pela heurística RFCC, a heurística RBCC também prioriza a eficiência de codificação.

Para que seja possível definir uma heurística desse tipo, foram utilizados como base os dados apresentados na subseção de avaliações preliminares, junto com os dados de redução de modos angulares e de tamanhos de bloco apresentados anteriormente nessa subseção.

Com o apoio das análises apresentadas, é perceptível que o impacto na eficiência de codificação é significativamente maior quando se remove tamanhos de bloco. Para que seja possível reduzir custo computacional sem perder muito desempenho na eficiência de codificação, a decisão da heurística RBCC foi remover uma pequena quantidade de tamanhos de bloco e remover uma grande quantidade de modos angulares.

Com base no que foi apresentado na Tabela 3, o candidato escolhido para a heurística RBCC foi o experimento M41. Com uma redução de mais de 61% dos modos angulares, ao custo de um BD-Rate de 0,52%. Com essa redução, a heurística usará apenas 26 modos angulares dos 67 disponíveis anteriormente. Os modos usados serão os modos: Planar, DC, 2, 3, 10, 17, 18, 19, 33, 34, 35, 36, 37, 38, 39, 42, 43, 46, 49, 50, 51, 54, 56, 64 e 65.

Na parte dos tamanhos de bloco, como pode ser visto na Tabela 4, o corte deve ser menor em porcentagem do que o corte feito nos modos angulares. Justamente devido ao impacto no BD-Rate ser bem maior quando tamanhos de bloco são descartados. Para essa heurística, o candidato escolhido será o B3. Esse experimento, conforme apresentado pela Tabela 4, possui um aumento no BD-Rate

de 1,81%, com uma redução de aproximadamente 17,65% no número de tamanhos de bloco avaliados. Os três tamanhos de bloco removidos são, segundo a Figura 12: 64x64, 32x32 e 32x16. Com isso, mantém-se os outros quatorze tamanhos de bloco (16x32, 16x16, 32x8, 8x32, 32x4, 4x32, 16x8, 8x16, 8x8, 16x4, 4x16, 8x4, 4x8 e 4x4). Entre modos angulares e tamanhos de bloco, a heurística RBCC apresenta uma redução total de aproximadamente 68,04% nas possibilidades de codificação de cada amostra do vídeo. Como explicado anteriormente na heurística RFCC, o VTM possui 1139 opções de codificação disponíveis para cada amostra do vídeo, enquanto a heurística RBCC possui apenas 364 opções de codificação disponíveis. Ou seja, a heurística RBCC utiliza apenas 31,96% das opções de codificação disponíveis originalmente pelo VTM.

Assim como na avaliação da heurística RFCC, os experimentos realizados para a heurística RBCC foram baseados no documento CTC, utilizando as especificações citadas anteriormente na parte de configuração experimental. Dois resultados foram considerados para avaliar o impacto dessa heurística: o BD-Rate e a RTC, do mesmo modo que na avaliação da heurística RFCC.

Os resultados de desempenho da heurística RBCC são apresentados na Tabela 5. O BD-Rate registrado pela heurística RBCC é de 2,17% em média, onde o intervalo de desempenho das classes de vídeo ficou entre 1,88% e 2,56%. Novamente é possível perceber através da Tabela 5 que os piores desempenhos no quesito eficiência de codificação ficaram com as classes de maior resolução. As classes A e B obtiveram os maiores BD-Rate (2,56% e 2,34% respectivamente). Os melhores valores foram presenciados nas classes C e D, classes com sequências de vídeo de menores resoluções (1,88% e 2,02% respectivamente). Quando o RTC da heurística é analisado, é possível observar uma diminuição na complexidade de 18,72% em média. De forma coerente com o resultado anterior, o RTC, é maior para sequências de vídeo de resoluções maiores, enquanto resoluções menores obtiveram menores reduções de custo computacional. É possível perceber que, quando as classes E e F são comparadas, a classe F apresentou melhores resultados de BD-Rate e de redução de custo computacional. O melhor resultado de RTC registrado na Tabela 5 foi para a classe A, com 22,97% de redução, enquanto o menor RTC é da classe C, com 14,31% de redução.

Quando as heurísticas RFCC e RBCC são comparadas, é importante ressaltar que o objetivo visado na heurística RBCC foi alcançado. De fato, a heurística RBCC possui um BD-Rate significativamente inferior ao da heurística RFCC (2,17% contra 8,62%), ou seja, o impacto na eficiência de codificação é quatro vezes menor. Naturalmente, e como esperado, essa melhora no BD-Rate trouxe um custo na redução do custo computacional (18,72% contra 60,92%), com uma RTC 3,3 vezes menor na RBCC.

Tabela 5 - Média de BD-Rate e RTC por classe de sequências de vídeo do CTC para heurística RBCC

Classe	Resolução	Quantidade de Sequências	BD-Rate	RTC
A	3840x2160	6	2,56%	22,97%
B	1920x1080	5	2,34%	22,45%
C	832x480	4	1,88%	14,31%
D	416x240	4	2,02%	15,75%
E	1280x720	3	2,15%	17,54%
F	832x480 até 1920x1080	4	2,07%	19,3%
Média			2,17%	18,72%

Fonte: Elaborada pelo autor (2024).

5 Implementações de Hardware

Nesse capítulo do trabalho serão apresentadas as implementações de todas as arquiteturas propostas. Como todas as propostas são voltadas aos modos angulares do VVC, as unidades de processamento dos modos angulares são as mesmas para todas as implementações e a diferença está no número de modos angulares suportados. Ainda nessa seção, será exposto o funcionamento de cada unidade de processamento dos modos angulares.

5.1 Parte Topo das Arquiteturas

Esse trabalho envolve três arquiteturas similares, porém com algumas diferenças em suas partes topo (ou parte de mais alto nível hierárquico). Em todos os casos, as arquiteturas foram projetadas para processar, em paralelo, todas as amostras de uma linha ou coluna do bloco a ser predito, independentemente do tamanho do bloco. Os tamanhos de bloco são processados de forma serial por todas as arquiteturas, embora o número de tamanhos de bloco varie de arquitetura para arquitetura. A quantidade de modos angulares processados varia para cada arquitetura.

A primeira arquitetura implementa todos os 67 modos angulares e todos os 17 tamanhos de bloco disponíveis no VVC (arquitetura Sem Redução de Custo Computacional - SRCC). Essa arquitetura foi desenvolvida para que fosse possível ter uma base de comparação para as demais arquiteturas e, também, para que fosse disponibilizada uma arquitetura sem perdas de eficiência de codificação, uma vez que todos os modos e todos os tamanhos de bloco são suportados. Neste caso, todos os 67 modos angulares são calculados em paralelo para cada tamanho de bloco. Com a arquitetura SRCC, é possível tanto mostrar o custo em hardware para a predição intra quadro completa do VVC, quanto evidenciar o impacto em redução de custo computacional que as outras arquiteturas apresentam.

A arquitetura RFCC, implementa a heurística RFCC, que foca em meios mais agressivos de redução de custo computacional para a criação de uma arquitetura com menor custo de hardware e menor consumo de energia. Assim apenas 18 modos angulares e cinco tamanhos de blocos quadrados são suportados. Nessa arquitetura

os modos angulares são calculados em duas etapas, com nove modos calculados em paralelo.

Por fim, a terceira arquitetura (arquitetura RBCC) implementa a heurística RBCC e procura atingir um resultado mais balanceado entre taxa de processamento e eficiência de codificação. Esta arquitetura tem como foco a redução do alto custo computacional, reduzindo modos angulares e tamanhos de bloco utilizados, da mesma forma que a arquitetura RFCC, porém em uma escala menor. Nesse caso, são suportados 26 modos angulares e 14 tamanhos de bloco, como já discutido. Neste caso, os 26 modos angulares são calculados em paralelo para cada tamanho de bloco. Essa redução de custo computacional menor serve para tornar a arquitetura RBCC mais competitiva em relação a eficiência de codificação em relação à arquitetura RFCC. As estratégias algorítmicas usadas nas arquiteturas RFCC e RBCC já foram explicadas na parte de desenvolvimento de heurística. O foco dessa seção é apresentar como as heurísticas foram projetadas a nível de hardware.

5.1.1 Arquitetura SRCC

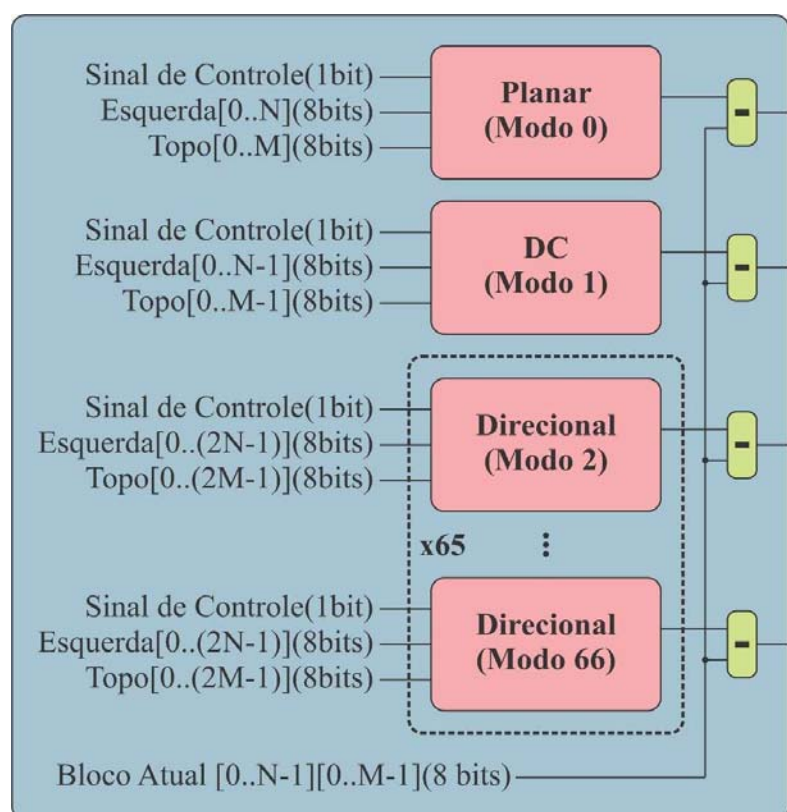
Como não existem trabalhos na literatura para uma justa comparação do impacto que as arquitetura RFCC e RBCC apresentam, foi realizada a implementação de uma arquitetura completa, contendo todos os modos angulares e todos os tamanhos de bloco da predição intra quadro do VVC (SRCC), afim de evidenciar de forma mais clara os pontos fortes e fracos das arquiteturas RFCC e RBCC e, também, para ter uma arquitetura disponível que não gere perdas de eficiência de codificação. Essa arquitetura foi aceita para publicação em (BORGES et al., 2024).

A Figura 13 apresenta o diagrama da parte topo da arquitetura SRCC. O design da arquitetura foi dividido em 67 unidades de processamento: a unidade Planar, a unidade DC e 65 unidades direcionais (uma unidade direcional para cada modo angular definido pelo VVC). Como esse trabalho foca em alta performance, todas as unidades trabalham de forma paralela.

Como cada unidade de processamento é responsável por um modo angular e todos os modos são calculados em paralelo, o número de ciclos para codificar todos os modos angulares presentes na arquitetura para cada tamanho de bloco é de $N + 1$, onde N representa as linhas de um tamanho de bloco $N \times M$. O cálculo de tempo é

baseado na unidade de processamento mais lenta, que é a unidade Planar, que será detalhada mais para frente neste capítulo e que necessita de dois ciclos de clock para finalizar seus cálculos (todas as demais unidades necessitam de apenas um ciclo). Considerando que o máximo valor de N ou M dos tamanhos de bloco suportados pela arquitetura SRCC é de 64 (máximo tamanho suportado pela predição intra quadro do VVC), então esta arquitetura necessita de 65 ciclos para processar um bloco, no pior caso. Considerando todos os 17 tamanhos de bloco suportados, com seus diferentes tamanhos, para processar uma CTU de 64x64 amostras, a arquitetura necessita de 8.645 ciclos para calcular todos os 67 modos de predição para todos os 17 tamanhos de bloco. Estes valores estão detalhados no próximo capítulo.

Figura 13 - Diagrama parte topo da arquitetura Sem Redução de Custo Computacional (SRCC) proposta para a predição intra quadro do VVC



Fonte: Elaborada pelo autor (2024).

A Figura 13 apresenta os valores de entrada das unidades de processamento. Cada unidade recebe como entrada um sinal de controle de um bit, que define quando o módulo está ou não ativo. Além disso, cada unidade de processamento recebe as

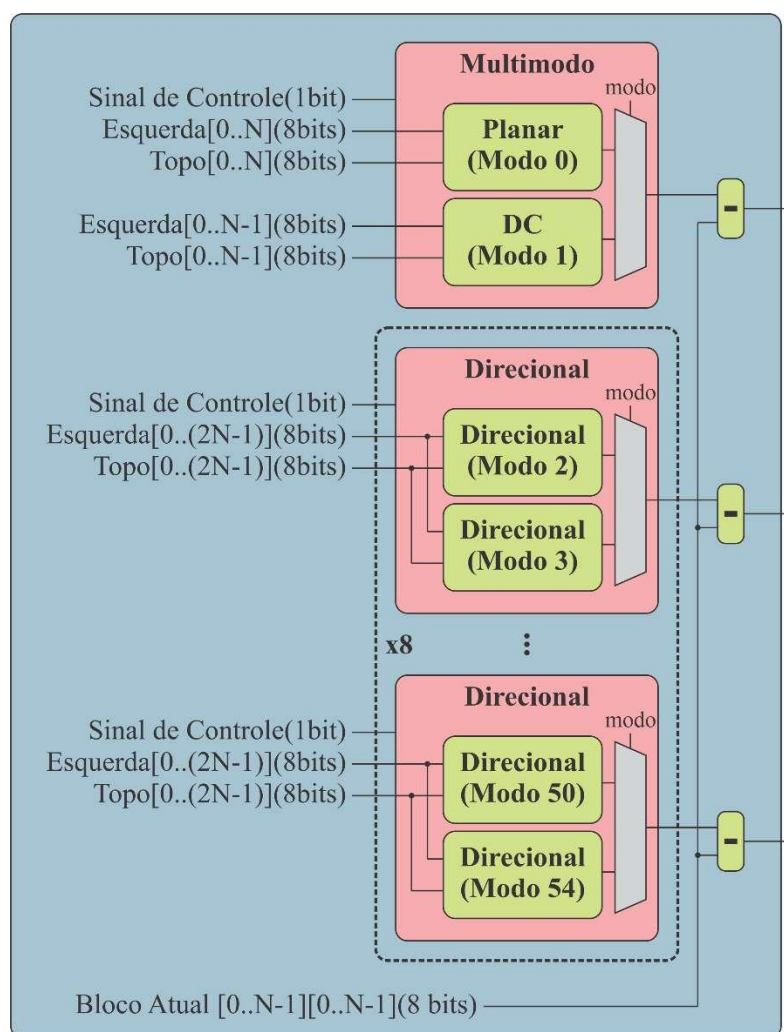
amostras vizinhas da Esquerda e do Topo, com 8 bits por amostra. Estas amostras são necessárias para a codificação do bloco a ser predito. O número de amostras de entrada é dependente do número de linhas (N) ou de colunas (M) do bloco a ser processado. Como representado na Figura 13, após a obtenção do bloco predito, o resíduo é obtido através da subtração feita com o bloco atual, também com amostras de 8 bits e de tamanho $N \times M$. As unidades de processamento Planar e DC possuem entradas diferentes das unidades direcionais devido à forma com que predizem o bloco ser diferente da forma de predição realizada pelas unidades direcionais. Cada unidade de processamento será explicada detalhadamente na seção de unidades de processamento de modos angulares.

5.1.2 Arquitetura RFCC

A arquitetura RFCC é uma arquitetura de hardware dedicada que implementa a heurística *hardware-friendly* RFCC para a predição intra quadro do VVC, focando em alta performance e baixo consumo energético. Essa arquitetura foi publicada em (BORGES et al., 2023). A heurística RFCC suporta 18 modos angulares e cinco tamanhos de bloco (apenas os tamanhos quadrados). Como pode ser visto na Figura 14, a arquitetura RFCC foi dividida em nove módulos: Um módulo chamado de multimodo e oito módulos direcionais. O módulo multimodo possui duas unidades de processamento específicas, são as unidades Planar e DC. Os módulos direcionais são responsáveis por duas unidades de processamento direcionais cada, totalizando 16 modos angulares. No caso da heurística RFCC, os modos direcionais disponíveis são: 2, 3, 7, 10, 18, 23, 26, 30, 33, 34, 35, 43, 46, 49, 50 e 54. Exemplificando, o primeiro módulo direcional é responsável pelas unidades direcionais que utilizam os modos angulares 2 e 3, o segundo módulo direcional é responsável pelos modos 7 e 10, e assim sucessivamente, até o último módulo, que é responsável pelos modos 50 e 54. Como cada módulo é responsável por duas unidades de processamento, o tempo de codificar todos os modos angulares presentes na arquitetura para cada tamanho de bloco é de $2N + 3$, onde N representa as linhas de um tamanho de bloco $N \times N$ (apenas blocos quadrados são suportados). O cálculo de tempo é baseado na unidade de processamento mais lenta, Planar, como já comentado. O máximo valor de N dos tamanhos de bloco suportados pela arquitetura RFCC também é de 64, uma

vez que todos os blocos quadrados são suportados (incluindo o bloco 64x64). Então esta arquitetura necessita de 133 ciclos para processar um bloco, no pior caso. Considerando todos os cinco tamanhos de bloco suportados, com seus diferentes tamanhos, a arquitetura necessita de 4.993 ciclos para calcular os 16 modos de predição para os cinco tamanhos de bloco suportados, novamente considerando uma CTU de 64x64 amostras. Estes valores estão detalhados no próximo capítulo.

Figura 14 - Diagrama parte topo da arquitetura de Redução Forte de Custo Computacional (RFCC) proposta para a predição intra quadro do VVC



Fonte: Elaborada pelo autor (2024).

A Figura 14 também apresenta os valores de entrada dos módulos. Cada módulo recebe como entrada um sinal de controle de um bit, que define quando o módulo está ou não ativo e quando está processando um modo ou outro. Além disso,

cada unidade de processamento recebe as amostras vizinhas da Esquerda e do Topo, necessárias para a codificação do bloco a ser predito, com 8 bits por amostra. Após gerar o bloco predito, o resíduo de cada modo é obtido. Esse resíduo é extraído através da subtração entre as amostras do bloco predito e do bloco original atual, representado na Figura 14 como bloco atual, também com 8 bits por amostra. As unidades de processamento serão explicadas detalhadamente na seção de unidades de processamento de modos angulares.

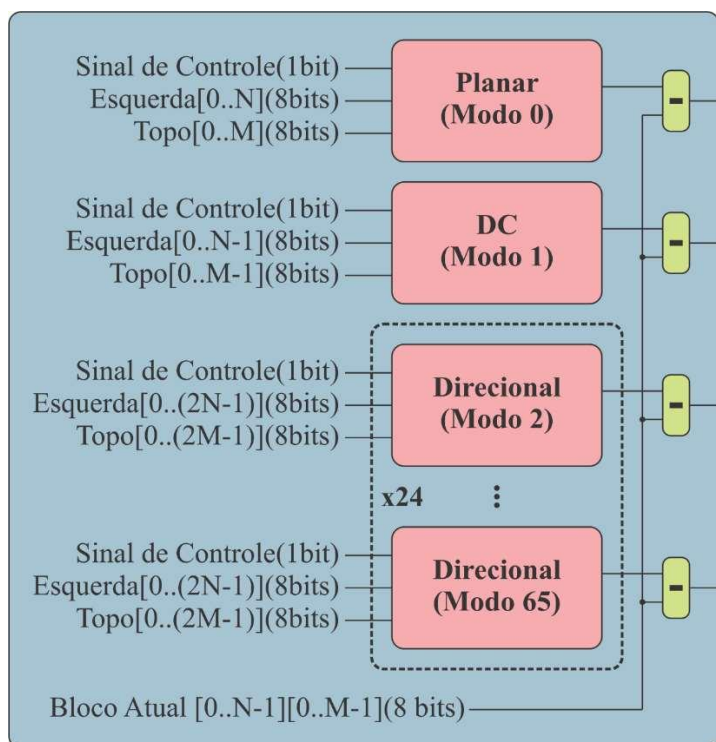
5.1.3 Arquitetura RBCC

A arquitetura RBCC é uma arquitetura criada com base na heurística RBCC discutida na seção anterior. Essa heurística tem como foco a redução de custo computacional, mas com uma baixa redução na eficiência de codificação. Como o foco deste trabalho é a criação de arquiteturas de hardware voltadas aos modos angulares da predição intra quadro do padrão VVC, e devido ao fato do custo computacional do padrão VVC ser significativamente elevado, foi necessária então a utilização da heurística RBCC como base para a criação da arquitetura RBCC. Lembrando que a heurística RBCC suporta 26 modos angulares e 14 tamanhos de bloco.

Assim como a arquitetura SRCC, a arquitetura RBCC é uma arquitetura de hardware de alta performance que suporta parte dos modos angulares (modos Planar, DC, 2, 3, 10, 17, 18, 19, 33, 34, 35, 36, 37, 38, 39, 42, 43, 46, 49, 50, 51, 54, 56, 58, 64 e 65) e a maioria dos tamanhos de bloco definidos para a predição intra quadro do padrão VVC (todos os tamanhos de bloco disponíveis para a predição intra quadro com exceção dos tamanhos 64x64, 32x32 e 32x16). A Figura 15 apresenta o diagrama da parte topo da arquitetura RBCC. O design da arquitetura foi dividido em 26 unidades de processamento: a unidade Planar, a unidade DC e 24 unidades direcionais (uma unidade direcional para cada modo angular definido pelo VVC). Como esse trabalho possui como foco a alta performance, todas as unidades trabalham de forma paralela. De forma idêntica à arquitetura SRCC, cada módulo é responsável por uma unidade de processamento, então o tempo de codificar todos os modos angulares presentes na arquitetura é de $N + 1$, onde N representa as linhas de um tamanho de bloco $N \times M$. O cálculo de tempo é baseado na unidade de

processamento mais lenta, que é a Planar, como já comentado. O máximo valor de N dos tamanhos de bloco suportados pela arquitetura RBCC é de 32 (nos blocos 32x8 e 32x4). Então esta arquitetura necessita de 33 ciclos para processar um bloco, no pior caso. Considerando todos os 14 tamanhos de bloco suportados, com seus diferentes tamanhos, a arquitetura necessita de 8.184 ciclos para calcular os 26 modos de predição para os 14 tamanhos de bloco suportados, considerando uma CTU de 64x64 amostras. Estes valores estão detalhados no próximo capítulo.

Figura 15 - Diagrama parte topo da arquitetura de Redução Balanceada de Custo Computacional (RBCC) proposta para a predição intra quadro do VVC



Fonte: Elaborada pelo autor (2024).

A Figura 15 apresenta os valores de entrada das unidades de processamento da arquitetura RBCC. Cada unidade recebe como entrada um sinal de controle de um bit, que define quando o módulo está ou não ativo. Além disso, cada unidade de processamento recebe as amostras vizinhas à Esquerda e as amostras vizinhas ao Topo, todas com 8 bits, necessárias para a codificação do bloco a ser predito. Como representado na Figura 15, após a obtenção do bloco predito, o resíduo é obtido através da subtração feita entre as amostras do bloco atual, também de 8 bits, e as amostras do bloco predito. As unidades de processamento Planar e DC possuem

entradas diferentes das unidades direcionais devido à forma com que predizem o bloco ser diferente da forma de predição realizada pelas unidades direcionais. Cada unidade de processamento será explicada detalhadamente na seção de unidades de processamento de modos angulares.

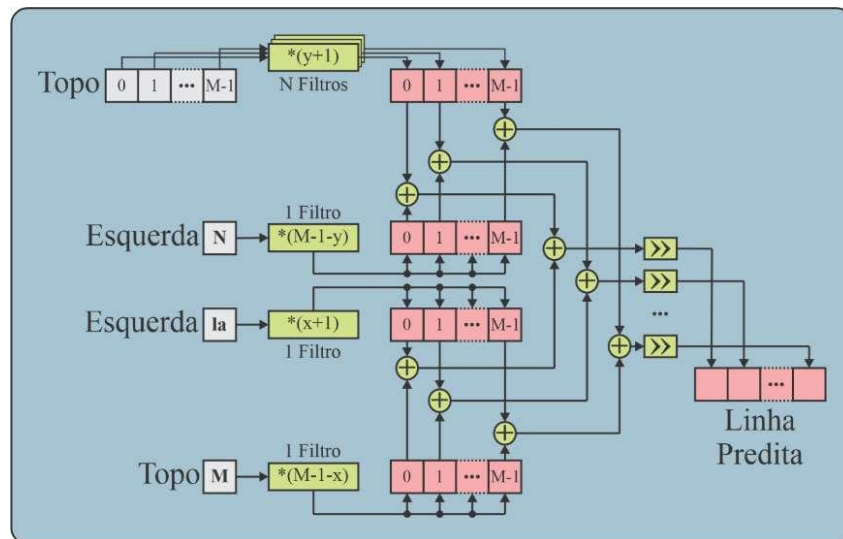
5.2 Unidades de Processamento dos Modos Angulares

5.2.1 Unidade Planar

Como pode ser visto na Figura 16, a unidade que representa o modo Planar mostra as operações realizadas para processar CUs de tamanho $N \times M$, onde N representa o número de linhas do bloco, enquanto M representa o número de colunas do bloco. Na Figura 16 ainda temos alguns símbolos como x e la , estes representam a posição da amostra e a linha atual, respectivamente.

A unidade Planar recebe como parâmetros de entrada: todas as amostras vizinhas que estão no topo do bloco a ser predito ($0 - M-1$), uma amostra vizinha à esquerda do bloco a ser predito na posição da linha atual (la) e uma amostra vizinha à esquerda do bloco a ser predito na posição N .

Figura 16 - Diagrama de bloco do modo Planar



Fonte: Elaborada pelo autor (2024).

Todas as unidades dos modos angulares foram projetadas para processar uma linha do bloco atual por vez, independentemente do número de amostras presentes na linha. No caso do modo Planar, são necessários dois ciclos para processar a primeira linha. No primeiro ciclo, todos os registradores intermediários são carregados com os valores dos respectivos filtros. No segundo, todas as acumulações são realizadas em paralelo, bem como os respectivos deslocamentos, para gerar a linha predita. A partir da segunda linha, um sistema de pipeline é utilizado para garantir o aproveitamento do hardware durante o processo de predição, assim o modo planar necessita de $N+1$ ciclos para prever um dado bloco $N \times M$.

Cada amostra de entrada passa por um filtro no primeiro momento. As amostras vizinhas ao topo da arquitetura no intervalo de 0 a $M-1$ passam pelo mesmo filtro. Esse filtro multiplica cada amostra pela soma entre a linha atual e um. O resultado é um vetor ponderado de tamanho M , onde cada posição do vetor será preenchida com o valor da posição respectiva do intervalo das amostras vizinhas multiplicado pelo filtro, como mostra a Figura 16. A amostra vizinha à esquerda na posição N passa pelo filtro que multiplica o valor da amostra pela subtração entre N , um e a linha atual. A amostra vizinha à esquerda na posição $1a$ passa pelo filtro que multiplica o valor da amostra pela soma de x com um. Por fim, a amostra vizinha ao topo na posição M passa pelo filtro que multiplica o valor da amostra pela subtração entre M , um e x (PFAFF et al., 2021). As duas amostras à esquerda e a amostra topo na posição M , quando passam pelo filtro, também produzem como resultado um vetor ponderado de tamanho M , porém, diferente das amostras topo do intervalo de 0 a $M-1$, essas replicam o resultado da multiplicação entre o valor presente em sua posição e seu respectivo filtro para todas as posições de seus vetores, como representado na Figura 16. Os valores dos filtros são salvos nos registradores apresentados em rosa na Figura 16. Em um único ciclo todos os registradores intermediários são preenchidos pelos valores dos filtros.

Após, esses registradores são somados com outros registradores de mesmo índice até termos um único registrador contendo o valor de todas as amostras multiplicadas pelos pesos dos filtros somados, como mostra a Figura 16. Por fim, esse valor total é dividido pela soma dos pesos dos filtros (essa divisão é feita através do deslocar presente ao final da Figura 16), gerando o valor final da amostra predita.

É importante ressaltar que as multiplicações citadas no parágrafo acima e apresentadas na Figura 16 foram substituídas por deslocadores e somadores, para melhorar o desempenho da arquitetura.

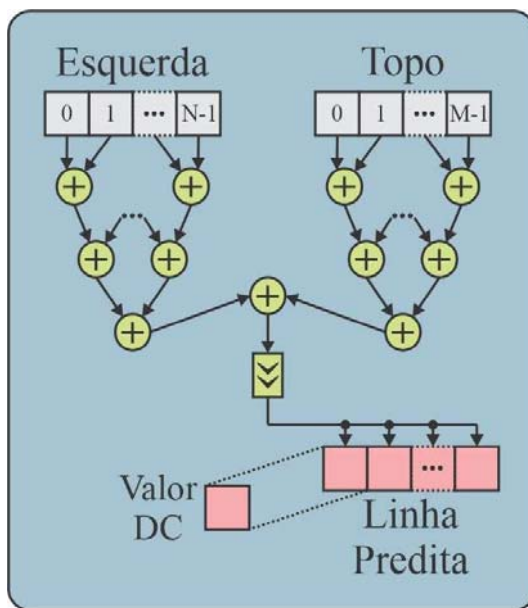
5.2.2 Unidade DC

A unidade DC, representada pela Figura 17, mostra as operações realizadas para processar CUs de tamanho $N \times M$, onde N representa o número de linhas do bloco, enquanto M representa o número de colunas do bloco. Esta arquitetura define o bloco a ser predito pelo valor médio de todas as amostras vizinhas. Esse valor médio é chamado de valor de DC (PFAFF et al., 2021).

O valor de DC é gerado pela soma de todas as amostras vizinhas, de duas a duas. Como pode ser visto nos somadores da Figura 17, todas as amostras da esquerda são somadas duas a duas, ao mesmo tempo em que é realizado o mesmo processo para as amostras topo. Esse processo se repete até chegarmos ao último somatório. O último somatório unifica o somatório das amostras da esquerda e o somatório das amostras topo, gerando o somatório total das amostras vizinhas. Após o somatório de todas as amostras, é realizado um deslocamento à direita a fim de obter o valor médio das amostras. O número de bits desse deslocamento varia de acordo com o tamanho do bloco e, para um bloco de tamanho $N \times M$, o somatório total será dividido pelo valor $N+M$, obtendo o valor de DC, que é usado para toda a linha predita.

Todo o processo de somatório de duas a duas amostras e deslocamento após o resultado do somatório é apresentado na Figura 17, especificamente na parte em verde. Por fim, o valor de DC é replicado para todo o bloco a ser predito (PFAFF et al., 2021). A implementação do modo DC processa uma linha por ciclo de clock. Para codificar uma CU de tamanho $N \times M$ serão necessários então N ciclos para todo o bloco.

Figura 17 - Diagrama de blocos do modo DC



Fonte: Elaborada pelo autor (2024).

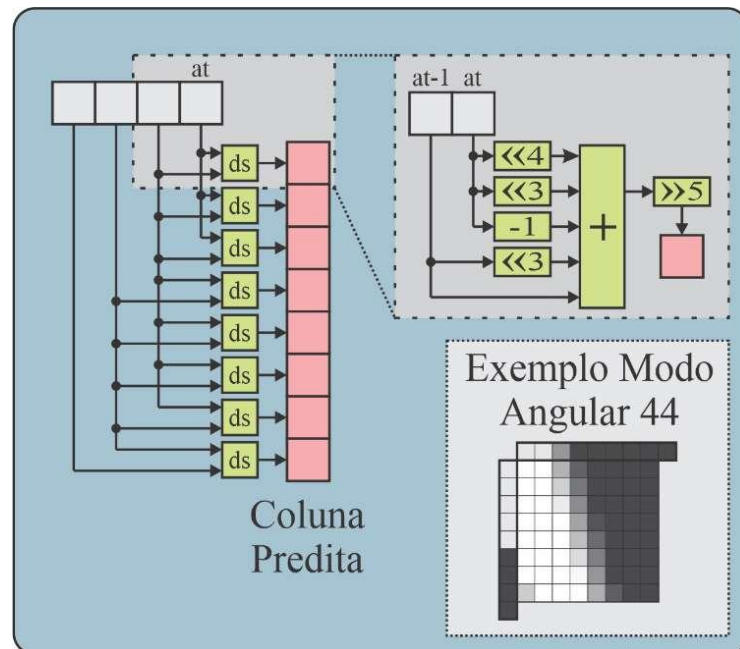
5.2.3 Unidades Direcionais

Na parte das unidades direcionais, cada modo angular usado possui uma unidade própria para processamento do mesmo. O processo é extremamente similar em todos os casos, alterando as amostras que serão utilizadas para geração do bloco a ser predito baseado no modo angular respectivo. A arquitetura para os modos horizontais (modos 2 a 34) realizam a predição do bloco linha a linha. Enquanto a arquitetura para os modos verticais (modos 35 a 66) realizam a predição do bloco coluna por coluna. Cada unidade direcional adota uma estratégia de derivação de um vetor de referência utilizando as amostras vizinhas. Baseada na direção do modo angular, somente a média ponderada de algumas amostras vizinhas serão utilizadas como base para a geração do bloco predito. Esse vetor de referência é usado para prever as amostras do bloco a ser predito (PFAFF et al., 2021).

A Figura 18 apresenta o processo para os modos verticais, utilizando como exemplo o modo angular 44 para prever um bloco de tamanho 8x8. Por ser um modo vertical, a predição é feita de coluna a coluna, como pode ser visto na Figura 18. Cada amostra predita da coluna é baseada em duas amostras vizinhas. Para processar uma coluna (ou linha, dependendo do modo angular), todos os componentes são necessários para realizar uma multiplicação de valores ponderados de amostras

vizinhas específicas. Esses pesos são fixos e estão representados na Tabela 1. Como pode ser visto na Figura 18, a amostra atual (at) está sendo multiplicada por 23 ($16*at + 8*at - at$), onde o $16*at$ é obtido por um deslocamento à esquerda de quatro bits no valor de at e o $8*at$ é obtido por deslocamento à esquerda de três bits no valor de at . Seguindo a mesma lógica, à esquerda da atual ($at-1$) está sendo multiplicada por 8. Após as multiplicações, é realizada a soma dos valores e, por fim, essa soma é dividida por 32 (deslocamento à esquerda de 5 bits) para gerar a amostra predita.

Figura 18 - Diagrama de blocos dos modos angulares, utilizando como exemplo o modo angular 44



Fonte: Elaborada pelo autor (2024).

Como pode ser notado pela própria Figura 18, os valores que multiplicam as amostras possuem valores fixos, e com isso, é possível substituir as multiplicações/divisões por somas e deslocamentos, para diminuir recursos de hardware, ampliar a taxa de processamento e reduzir o consumo energético da arquitetura.

As unidades direcionais dependem do tamanho do bloco para obter o número de ciclos necessários. Então, considerando um bloco $N \times M$, os modos angulares horizontais necessitam de N ciclos para gerar a predição do bloco de entrada, enquanto os modos angulares verticais necessitam de M ciclos para gerar a predição.

É importante ressaltar que o processo citado acima pela Figura 18, é o mesmo para qualquer tamanho de bloco e modo angular de predição. Ou seja, o tamanho da linha ou coluna a ser predita vai depender precisamente do tamanho de bloco que está sendo processado e do ângulo do modo que está sendo avaliado no momento.

6 Resultados e Discussão

As arquiteturas propostas nesse trabalho foram descritas em VHDL e as sínteses foram realizadas usando a ferramenta *RTL Compiler* da Cadence, visando a tecnologia TSMC 40nm. Os resultados de consumo energético consideram o chaveamento padrão definido pela ferramenta de síntese.

Para realização das sínteses foi utilizado um servidor com dois processadores da série Xeon Gold 6430, com processamento de 2.10 GHz cada. Cada processador consiste em 32 núcleos físicos e 64 threads. O servidor é equipado com 4 pentes de memória RAM AMD DIMM de 32 GB cada, com uma frequência de 4.400 MHz, totalizando 128 GB de memória RAM. A versão do sistema operacional presente no servidor é o Linux Ubuntu na versão 20.04.6.

Por conta do número de ciclos que as arquiteturas precisam para processar os quadros dos vídeos, para uma dada resolução, as frequências alvo serão naturalmente diferentes. Neste trabalho foi considerada a resolução alvo de 1920x1080 pixels, também chamada de Full HD ou FHD, com uma taxa de 30 quadros por segundo (qps). A frequência de operação de cada arquitetura tendo esta resolução alvo foi definida a partir de uma análise detalhada do número de ciclos necessários em cada caso.

A Tabela 6 apresenta esta análise. Esta tabela apresenta, para cada arquitetura, o número de ciclos necessários para processar cada tamanho de bloco e o número de ciclos necessários para processar todos os blocos daquele tamanho que cabem dentro de uma CTU de 64x64 amostras. Ao final da Tabela, também é apresentado o número total de ciclos necessários para processar cada CTU, para cada arquitetura. Lembrando que cada CTU pode conter um número diverso de blocos. Por exemplo, uma CTU de 64x64 amostras pode conter um bloco 64x64, quatro blocos 32x32, oito blocos 16x32 e assim por diante.

Em todas as arquiteturas foi considerado o pior caso (modo planar). No caso da arquitetura RFCC, são necessários $2N + 3$ ciclos para processar uma CU de tamanho $N \times N$ (todos os tamanhos de bloco utilizados na arquitetura RFCC são quadrados), como já discutido. Já as arquiteturas SRCC e RBCC possuem a mesma estratégia de processar os modos angulares em paralelo e, nestes casos, são necessários $N + 1$ ciclos para processar um tamanho de bloco $N \times M$.

Como pode ser visto na Tabela 6, a arquitetura RFCC é a arquitetura que necessita de menos ciclos para gerar uma CTU predita, justamente pelo fato de utilizar a menor quantidade de tamanhos de bloco. As arquiteturas SRCC e RBCC necessitam da mesma quantidade de ciclos para processar cada tamanho de bloco, mas a arquitetura SRCC requer um número maior de ciclos para processar toda a CTU, pois suporta três tamanhos de bloco a mais do que a arquitetura RBCC.

Tabela 6 - Tabela de ciclos de clock para cada CU e para cada CTU das arquiteturas RFCC, SRCC e RBCC

	RFCC		SRCC		RBCC	
	CU	CTU	CU	CTU	CU	CTU
64x64	131	131	65	65	-	-
32x32	67	268	33	132	-	-
32x16	-	-	33	264	-	-
16x32	-	-	17	136	17	136
16x16	35	560	17	272	17	272
32x8	-	-	33	528	33	528
8x32	-	-	9	144	9	144
32x4	-	-	33	1056	33	1056
4x32	-	-	5	160	5	160
16x8	-	-	17	544	17	544
8x16	-	-	9	288	9	288
8x8	19	1216	9	576	9	576
16x4	-	-	17	1088	17	1088
4x16	-	-	5	320	5	320
8x4	-	-	9	1152	9	1152
4x8	-	-	5	640	5	640
4x4	11	2816	5	1280	5	1280
Total	-	4991	-	8645	-	8184

Fonte: Elaborada pelo autor (2024).

A partir do número de ciclos necessários para processar uma CTU, é possível definir a frequência de operação para a resolução alvo (FHD a 30qps). Inicialmente é necessário calcular o número de CTUs que cabem em um quadro FHD. Então basta multiplicar esse número pelo número de ciclos necessários para processar uma CTU em cada uma das arquiteturas. A seguir basta multiplicar esse número por 30, uma vez que é necessário processar 30 quadros FHD por segundo.

Os resultados de síntese estão apresentados na Tabela 7. Naturalmente, a arquitetura SRCC é a que exigiu a maior área, uma vez que é a arquitetura que suporta o maior número de modos angulares e tamanhos de bloco. No extremo oposto está a arquitetura RFCC, que possui a maior restrição no número de tamanhos de blocos e modos angulares. Também seguindo essa lógica, a arquitetura RBCC apresenta resultados intermediários de área, uma vez que é baseada em uma heurística com um número intermediário de tamanhos de bloco e modos angulares.

Tabela 7 - Resultados de síntese

	Arquitetura		
	SRCC	RFCC	RBCC
Área(k portas)	13.508,4	1.453	4.838,7
Frequência para FHD@30qps	131,3 MHz	75,8 MHz	124,3 MHz
Potência para FHD@30qps	755,18 mW	91,65 mW	270,5 mW

Fonte: Elaborada pelo autor (2024).

A partir dos resultados da Tabela 7 é possível observar que a arquitetura RFCC é a que exige a menor frequência de operação para processar vídeos FHD a 30qps, sendo quase a metade da frequência necessária para as demais arquiteturas. Esse resultado é natural, uma vez que a arquitetura RFCC necessita de um número muito inferior de ciclos para processar o vídeo, conforme já discutido. As arquiteturas SRCC e RBCC requerem frequências similares, uma vez que também demandam um número similar de ciclos para processar os vídeos.

Mas o resultado mais interessante é o que apresenta a potência dissipada pelas arquiteturas. Com uma frequência alvo muito inferior e com um hardware muito menor, a arquitetura RFCC apresenta resultados de potência mais do que oito vezes menores do que os resultados da arquitetura SRCC e quase três vezes menores do que a arquitetura RBCC.

É interessante perceber também que, como esperado, os resultados de síntese acompanham os resultados de eficiência de codificação. A arquitetura RFCC possui 8,62% de BD-Rate, ocupa 9,3 vezes menos área e dissipa uma potência 8,2 vezes menor do que a arquitetura SRCC, que tem 0% de BD-Rate. Por outro lado, a arquitetura RBCC, com um BD-Rate de 2,17%, ocupa uma área 2,8 vezes menor e dissipa uma potência também 2,8 vezes menor que a arquitetura SRCC. Esses resultados demonstram que as simplificações propostas são eficientes, mas que trazem um custo esperado em eficiência de codificação.

Como os trabalhos relacionados possuem diferentes padrões de codificação, diferentes tecnologias e avaliaram seus experimentos de forma diferente, uma justa comparação entre os trabalhos relacionados e as arquiteturas propostas por essa dissertação torna-se algo inviável. Dentre os trabalhos relacionados, apenas o trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) utiliza o mesmo padrão de codificação que as arquiteturas propostas, contudo a tecnologia utilizada por (AZGIN; KALALI; HAMZAOGLU, 2019) não é a mesma que os trabalhos propostos, além de não possuir resultados de consumo energético e eficiência de codificação, impedindo qualquer comparação relevante. A comparação de cada arquitetura se torna mais interessante entre as arquiteturas propostas justamente por estarem utilizando o mesmo padrão de codificação e a mesma tecnologia para desenvolvimento do hardware.

Como pode ser visto na Tabela 8, o trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) e a arquitetura SRCC são os trabalhos que suportam a maior quantidade de modos angulares dentre todos os trabalhos comparados (65 e 67 modos, respectivamente). Os trabalhos de (HUANG et al., 2016) e (ZHANG; LU, 2019) ambos suportam apenas 35 modos angulares devido a limitação do padrão de codificação HEVC. Enquanto o VVC possui 67 modos angulares ao todo, o HEVC possui apenas 35 modos angulares, como já discutido. As arquiteturas RFCC e RBCC, embora utilizem o VVC como padrão de codificação, ambas tiveram uma redução significativa no número de modos angulares utilizados. Enquanto a arquitetura RFCC utiliza cerca de 26,87% dos modos disponíveis, a arquitetura RBCC utiliza aproximadamente 38,81% dos modos disponíveis no padrão VVC. Como o foco são as arquiteturas de hardware, as arquiteturas RFCC e RBCC reduzem o custo computacional de forma a gerar melhores resultados em questão de área e consumo

energético. Os resultados relacionados ao hardware dessas arquiteturas serão discutidos a seguir.

Tabela 8 - Comparações com trabalhos relacionados

	(HUANG et al., 2016)	(ZHANG; LU, 2019)	(AZGIN; KALALI; HAMZA OGLU, 2019)	RFCC	SRCC	RBCC
Padrão	HEVC	HEVC	VVC	VVC	VVC	VVC
Modos	35	35	65	18	67	26
Tamanhos de bloco	4	4	4	5	17	14
BD-Rate	4,3%*	3,02%*	-	8,62%	0%	2,17%
Tecnologia	CMOS 55nm	TSMC 90nm	FPGA Virtex 7 28nm	TSMC 40nm	TSMC 40nm	TSMC 40 nm
Área(k portas)	1.571,7	2.288	-	1.453	13.508,4	4.838,7
Frequência para 1080p@30qp/s	147 MHz**	80 MHz**	119 MHz	75,8 MHz	131,3 MHz	124,3 MHz
Consumo para 1080p@30qp/s	97 mW**	59 mW**	-	91,65 mW	755,18 mW	270,5 mW

Fonte: Elaborada pelo autor (2024).

Legenda: * Resultados baseados no HEVC – metade da eficiência de codificação do VVC. ** Valores estimados.

Em relação à quantidade de tamanhos de bloco suportados por cada trabalho, os trabalhos de (HUANG et al., 2016) e (ZHANG; LU, 2019) suportam todos os tamanhos de bloco disponíveis para a predição intra quadro do padrão HEVC, que são apenas quatro (32x32, 16x16, 8x8 e 4x4). O restante dos trabalhos, por focarem no padrão VVC, suportam um número de tamanhos de bloco muito maior que os trabalhos baseados no HEVC. Dentre os trabalhos baseados no VVC, a arquitetura SRCC é a única que suporta todos os 17 tamanhos de bloco disponíveis para predição intra quadro.

O trabalho de (AZGIN; KALALI; HAMZA OGLU, 2019), apesar de utilizar o padrão VVC como base, suporta o mesmo número de tamanhos de bloco suportados pelos trabalhos de (HUANG et al., 2016) e (ZHANG; LU, 2019), com foco no HEVC. A arquitetura RFCC, com foco em redução forte de custo computacional, utiliza apenas

os cinco tamanhos de bloco quadrados definidos pelo VVC. Dentre os trabalhos apresentados com foco no VVC, (AZGIN; KALALI; HAMZAOGLU, 2019) e RFCC são aqueles com menor quantidade de tamanhos de bloco suportados. Por fim, a arquitetura RBCC junto com a arquitetura SRCC são as soluções com maior quantidade de tamanhos de bloco suportados, com 14 e 17 blocos, respectivamente.

O próximo tópico abordado pela Tabela 8 é o impacto na eficiência de codificação, representado pelo aumento no BD-Rate. Dentre os trabalhos relacionados, o trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) é o único que não apresenta um valor de BD-Rate gerado pelo trabalho, o que torna impossível este trabalho ser analisado no quesito eficiência de codificação. O trabalho de (HUANG et al., 2016), segundo a Tabela 8, apresentou um aumento no BD-Rate de 4,3%, enquanto o trabalho de (ZHANG; LU, 2019) apresentou um aumento no BD-Rate de 3,02%. É importante ressaltar que, apesar dos valores de BD-Rate apresentados pelos trabalhos de (HUANG et al., 2016) e (ZHANG; LU, 2019), ambos trabalhos são baseados no padrão de codificação HEVC. Nesse sentido, por possuir metade da eficiência de codificação que o padrão VVC, seria possível inferir que, se as soluções fossem aplicadas ao VVC, as perdas de eficiência de codificação seriam pelo menos o dobro da apresentada para o HEVC. De todo modo, é difícil realizar uma comparação justa entre os trabalhos com o padrão HEVC e os trabalhos com o VVC.

Os resultados de BD-Rate restantes são das arquiteturas apresentadas nessa dissertação. A arquitetura SRCC se destaca por ser a arquitetura com menor perda de eficiência de codificação dentre todos os trabalhos citados na Tabela 8. Por não remover nenhum modo angular ou tamanho de bloco, a arquitetura SRCC não apresenta perdas na eficiência de codificação (BD-Rate de 0%). Por outro lado, a arquitetura RFCC apresenta a maior perda na eficiência de codificação de todos os trabalhos presentes (um aumento de BD-Rate de 8,62%). Esse valor elevado se deve justamente pela remoção significativa de diversos modos angulares e tamanhos de bloco, visando o ganho em custo computacional que será apresentado mais à frente. A última arquitetura apresentada na Tabela 8, a arquitetura RBCC, possui um aumento no BD-Rate de 2,17%, sendo a segunda arquitetura com menor perda de eficiência de codificação.

Segundo a Tabela 8, todos os trabalhos relacionados, com exceção das arquiteturas apresentadas nessa dissertação, focam em tecnologias diferentes para

gerar os resultados de síntese de hardware. Essa discrepância de tecnologias é outro ponto muito importante que dificulta uma comparação justa entre os trabalhos que estão sendo apresentados. (HUANG et al., 2016), (ZHANG; LU, 2019) e (AZGIN; KALALI; HAMZAOGLU, 2019) utilizam tecnologias diferentes (CMOS 55nm, TSMC 90nm e FPGA Virtex 7 28nm, respectivamente). As arquiteturas SRCC, RFCC e RBCC, embora diferentes dos trabalhos citados anteriormente, utilizam a mesma tecnologia TSMC 40nm, permitindo uma comparação mais precisa entre as arquiteturas propostas por essa dissertação.

O resultado de área necessária para projetar cada uma das arquiteturas também é demonstrado na Tabela 8. Os valores de portas presentes na Tabela 8, referem-se a portas do tipo NAND2 equivalentes. Para projetar o trabalho de (HUANG et al., 2016), foi necessário uma área de 1.571,7 k portas, uma área 31,3% menor que a área necessária para o trabalho de (ZHANG; LU, 2019) (2.288 k portas). Porém, novamente cabe destacar que esses trabalhos focam no padrão HEVC. Esses trabalhos se mantêm como a primeira (trabalho de (HUANG et al., 2016)) e a segunda (trabalho de (ZHANG; LU, 2019)) menor área apresentada dentre todos os trabalhos presentes.

A arquitetura RFCC, com seu foco em redução significativa de complexidade computacional, mesmo sendo a arquitetura com maior perda na eficiência de codificação, cumpre seu objetivo, sendo um dos trabalhos com menor área necessária para o projeto de hardware (1.453 k portas). Por outro lado, segundo a Tabela 8, a arquitetura SRCC com foco em eficiência de codificação, paga o preço em custo computacional. A arquitetura SRCC demonstra ser a arquitetura com a maior área necessária para seu projeto de hardware, necessitando de 13.508,4 k portas, uma área cerca de 9,3 vezes maior que a área necessária na arquitetura RFCC.

A terceira arquitetura proposta por essa dissertação, a arquitetura RBCC, se apresenta como uma versão intermediária entre as arquiteturas RFCC e SRCC. RBCC necessita de uma área de 4.838,7 k portas para seu projeto de hardware. Essa área representa um valor de aproximadamente 3,33 vezes maior que a área apresentada por RFCC e cerca de 35,82% da área apresentada por SRCC. Com uma redução menor no custo computacional, é de se esperar que RBCC não atinja resultados tão promissores que RFCC quando a questão de área é analisada. Porém, a arquitetura RBCC se torna muito mais atrativa por possuir uma perda na eficiência

de codificação significativamente menor que RFCC, sem a necessidade do custo relativamente elevado da SRCC.

O trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) é focado em FPGA, então a comparação com área de hardware fica inviável, pois os dados são apresentados em termos de LUTs (GUPTA, 2023), blocos de memórias de acesso randômico (do inglês *Block Random Access Memory* - BRAM)(RUSSELL, 2022) e DFFs (STORR, 2013). Até existem estudos que tentam gerar métricas comparativas entre LUTs e portas equivalentes, como (“Comparing ASIC gate-equivalent with XU LUTs”, [s.d.]), mas como o trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) também usa BRAMs e DFFs, não é possível fazer uma conversão justa destes resultados para portas equivalentes. A seguir, a Tabela 8 apresenta os resultados de frequência gerados pelas arquiteturas para processar vídeos FHD em 30 quadros por segundo. Dos trabalhos sobre o codificador HEVC, o trabalho de (HUANG et al., 2016) apresenta uma frequência de 147 MHz, enquanto o trabalho de (ZHANG; LU, 2019) apresenta uma frequência de 80 MHz. É importante destacar que esses resultados são estimativas, pois os trabalhos listados acima não avaliaram suas arquiteturas para a resolução 1080p@30qps.

O trabalho de (AZGIN; KALALI; HAMZAOGLU, 2019) registra uma frequência de 119 MHz. A arquitetura RFCC apresenta a menor frequência de todos os trabalhos presentes na Tabela 8, com 75,8 MHz. A arquitetura SRCC apresenta uma frequência maior que a frequência apresentada por (AZGIN; KALALI; HAMZAOGLU, 2019), com o valor de 131,3 MHz, a maior frequência apresentada na Tabela 8. A arquitetura RBCC apresenta uma frequência de aproximadamente 124,3 MHz para a resolução 1080p@30qps.

Por fim, a Tabela 8 apresenta a dissipação de potência de cada trabalho comparado para a resolução 1080p@30qps. Dentre os trabalhos relacionados ao codificador HEVC, o trabalho de (ZHANG; LU, 2019) apresenta a menor dissipação de potência, com o valor de 59 mW. Já o trabalho de (HUANG et al., 2016) apresenta uma dissipação de potência de 97 mW. Mesmo focando em um padrão de codificação com custo computacional muito menor, esses trabalhos estão entre os trabalhos com menor dissipação de potência. Infelizmente, é impossível comparar o valor de dissipação de potência da arquitetura de (AZGIN; KALALI; HAMZAOGLU, 2019), pois este trabalho não apresenta resultados de dissipação de potência em seu artigo.

A arquitetura RFCC novamente cumpre seu objetivo quando seu resultado de dissipação de potência é analisado. Esta arquitetura, dentre as arquiteturas baseadas no VVC, é a que possui a menor dissipação de potência (91,65 mW para a resolução 1080p@30qps). Por outro lado, a arquitetura SRCC, com seu custo elevado para manter a eficiência de codificação, apresenta uma dissipação de potência muito elevada, chegando a 755,18 mW. Essa é a arquitetura com a maior dissipação de potência dentre todas as arquiteturas apresentadas na Tabela 8.

Em uma comparação de dissipação de potência entre as arquiteturas SRCC e RFCC, a arquitetura SRCC possui uma dissipação de potência 8,24 vezes maior do que a arquitetura RFCC. A arquitetura RBCC por sua vez, possui uma dissipação de potência de 270,5 mW, tornando-se a segunda arquitetura com menor dissipação de potência dentre as arquiteturas baseadas no VVC. A arquitetura RBCC, apesar de possuir uma dissipação de potência cerca de 2,95 vezes maior do que a arquitetura RFCC, também apresenta uma dissipação de potência de apenas 35,82% dissipação de potência requerida pela arquitetura SRCC.

7 Considerações Finais

Com a enorme demanda de vídeos digitais para diversos ramos da sociedade, a tecnologia evolui em ritmo acelerado para suprir esse consumo de vídeos digitais com qualidade cada vez maior. Com isso, os padrões de codificação de vídeo atuais foram criados para diminuir os tamanhos significativos de vídeos digitais sem perdas na eficiência de codificação. Porém, um custo computacional significativamente elevado é necessário para atingir esse objetivo. Como forma de melhorar esses codificadores, diversos pesquisadores por todo mundo realizam pesquisas em ferramentas específicas desses codificadores, a fim de melhorar seu desempenho ou diminuir seu custo computacional.

O presente trabalho propôs duas heurísticas em software e três arquiteturas de hardware com o objetivo de apresentar diferentes opções para a ferramenta de modos angulares da predição intra quadro do padrão de codificação VVC. Além disso, uma análise comparativa detalhada foi realizada, com o objetivo de entender o impacto em software e hardware que cada metodologia apresentada nesse trabalho teve para o codificador VVC.

Com o atual momento, em que muitas pesquisas estão sendo feitas com foco em utilização de inteligência artificial, trabalhos voltados a parte de hardware estão mais escassos devido à dificuldade e custo elevado de adicionar heurísticas de inteligência artificial à projetos de hardware. Com esse fato, é difícil a comparação de trabalhos em nível de hardware com o trabalho proposto por essa dissertação devido à falta destes na literatura. De mesmo modo, esses trabalhos utilizando inteligência artificial no codificador VVC para a predição intra quadro são, em grande maioria, focados em melhorias no particionamento de tamanhos de bloco e decisão de utilização de determinada ferramenta presente na predição intra quadro em determinada circunstância. Tendo isso em vista, trabalhos como o apresentado nessa dissertação, com o foco em melhorias em uma ferramenta específica da predição intra quadro do VVC são praticamente inexistentes na literatura atual.

Pelos fatos listados logo acima, este trabalho teve como foco não só a criação de uma arquitetura de hardware baseada em uma heurística realizada, como a demonstração de diversas opções de arquiteturas de hardware para a ferramenta modos angulares da predição intra quadro do codificador VVC. Com a criação de duas heurísticas e três arquiteturas de hardware, foi possível não só realizar uma análise

mais detalhada do impacto de cada método adotado por este trabalho, como entender em que circunstâncias um determinado método tem mais relevância que os outros.

A primeira heurística apresentada por essa dissertação é a heurística RFCC. A heurística possui como foco a redução significativa do custo computacional presente na ferramenta modos angulares da predição intra quadro do VVC. Os focos de redução da heurística foram nos modos angulares e nos tamanhos de bloco disponíveis pelo VVC. Os resultados de avaliação da heurística no software de referência VTM indicaram uma redução de 60,92% no tempo de codificação. Porém essa redução expressiva veio ao custo do elevado aumento no BD-Rate de aproximadamente 8,62%, o que indica uma significativa perda na eficiência de codificação.

Uma arquitetura de hardware foi implementada utilizando a heurística RFCC como base, sendo nomeada de arquitetura RFCC. Essa arquitetura divide os 18 modos angulares da heurística RFCC em nove unidades de processamento paralelas, onde cada unidade de processamento avalia dois modos angulares um após o outro. Os resultados da arquitetura RFCC demonstram que a heurística RFCC, de drástica redução de custo computacional para implementação em hardware, cumpre seu papel. A arquitetura RFCC foi a que apresentou uma das menores áreas requeridas para implementação de hardware e a menor dissipação de potência dentre todos os trabalhos relacionados nessa avaliação.

A segunda heurística foi batizada de RBCC. Essa heurística teve como objetivo trazer redução de custo computacional sem tanto impacto na eficiência de codificação.

A heurística RBCC remove de forma mais significativa os modos angulares disponíveis enquanto realiza uma pequena redução nos tamanhos de bloco disponíveis. Isso se deve justamente pela análise dos testes realizados no VTM apontarem que a remoção de tamanhos de bloco tem um impacto maior na eficiência de codificação do que a remoção de modos angulares. Os resultados de avaliação da heurística no software de referência VTM apresentam uma redução de 18,72% no tempo de codificação. Uma redução não tão expressiva quanto a redução apresentada pela heurística RFCC, entretanto a heurística RBCC se demonstra mais competitiva na questão da eficiência de codificação, onde os resultados indicam uma perda de 2,17% no BD-Rate.

A arquitetura RBCC, baseada na heurística RBCC, divide seus 26 modos angulares disponíveis em uma unidade de processamento para cada modo angular, totalizando 26 unidades de processamento operando em paralelo. Os resultados gerados pela arquitetura RBCC provam que uma redução de complexidade mais eficiente pode gerar resultados promissores tanto em software, quanto em hardware, mesmo não sendo os melhores resultados. Os resultados de área, dissipação de potência e eficiência de codificação se mostraram próximos aos melhores resultados apresentados nessa dissertação.

Por fim, esta dissertação também apresentou uma arquitetura de hardware que utiliza todo o potencial disponibilizado para a ferramenta modos angulares do VVC, que foi batizada de SRCC. Por ser uma representação em hardware da ferramenta modos angulares, essa arquitetura não implica em perdas na eficiência de codificação. Essa arquitetura se torna importante para dimensionar o impacto que uma ferramenta como os modos angulares pode ter em hardware quando implementada por completo.

A arquitetura SRCC, do mesmo modo que a arquitetura RBCC, utilizou uma unidade de processamento para cada modo angular suportado. Nesse caso, foram 67 unidades de processamento trabalhando em paralelo. Os resultados de hardware da arquitetura SRCC provam a dificuldade de implementação física dessa ferramenta. A arquitetura SRCC demonstrou possuir a maior área requerida para o projeto de hardware, além de possuir a maior dissipação de potência dentre os trabalhos relacionados, valores esses que são distantes dos resultados gerados pelos outros trabalhos.

É importante destacar que cada solução proposta nessa dissertação pode ser a melhor opção a ser utilizada, dependendo do contexto em que estas são necessárias. Se existe uma limitação de hardware e a eficiência de codificação não tem tanta prioridade, a melhor opção pode ser a arquitetura RFCC. Esta arquitetura com redução significativa de custo computacional, facilita o projeto do hardware, mas traz perdas significativas na eficiência de codificação. Se o cenário exige máxima eficiência de codificação e o custo do hardware não é um problema, então a arquitetura SRCC pode ser a melhor opção. Por fim, se o cenário apresenta certas limitações nos custos do hardware e, também, nas perdas aceitáveis de eficiência de codificação, a arquitetura RBCC pode ser a melhor opção.

Em uma perspectiva de futuro, existem algumas possibilidades que poderiam complementar o presente trabalho. Por exemplo, poderiam ser desenvolvidas heurísticas voltadas às outras ferramentas da predição intra quadro do VVC, suportando o desenvolvimento de arquiteturas de hardware baseadas nessas heurísticas.

Indo de encontro as tendências atuais, poderiam ser realizadas pesquisas utilizando inteligência artificial, porém com foco em implementações de hardware. Onde o objetivo seja não só obter ganhos em redução computacional e/ou eficiência de codificação, mas também pensar em estratégias com inteligência artificial que sejam possíveis de serem implementadas em hardware. A geração de resultados de hardware com inteligência artificial poderia evidenciar de forma mais concreta a viabilidade e o impacto desses dois tópicos juntos, uma vez que não existem trabalhos que sequer apontam tentativas de utilizar ambos na literatura.

Referências

- AGOSTINI, L. V. Desenvolvimento de Arquiteturas de Alto Desempenho dedicadas à compressão de vídeo segundo o Padrão H.264/AVC. [s.l: s.n.].
- AZGIN, H.; KALALI, E.; HAMZAOGLU, I. An Efficient FPGA Implementation of Versatile Video Coding Intra Prediction. 2019 22nd Euromicro Conference on Digital System Design (DSD). Anais... Em: 2019 22ND EUROMICRO CONFERENCE ON DIGITAL SYSTEM DESIGN (DSD). ago. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8875183>>. Acesso em: 12 fev. 2024
- BHASKARAN, V.; KONSTANTINIDES, K. Image and Video Compression Standards: Algorithms and Architectures. [s.l.] Springer Science & Business Media, 1997.
- BJONTEGAARD, G. Improvements of the BD-PSNR model. jul. 2008.
- BORGES, V. et al. Efficient Architecture for VVC Angular Intra Prediction based on a Hardware-Friendly Heuristic. 2023 IEEE 14th Latin America Symposium on Circuits and Systems (LASCAS). Anais... Em: 2023 IEEE 14TH LATIN AMERICA SYMPOSIUM ON CIRCUITS AND SYSTEMS (LASCAS). fev. 2023. Disponível em: <<https://ieeexplore.ieee.org/document/10108393>>. Acesso em: 12 fev. 2024
- BORGES, V. et al. High-Throughput Hardware Design for the Complete VVC Angular Intra Prediction. 2024 IEEE International Conference on Electronics, Circuits, and Systems (ICECS). Anais... Em: 2024 IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS, AND SYSTEMS (ICECS). nov. 2024.
- BOSEN, F. et al. JVET common test conditions and software reference configurations for SDR video. Joint Video Experts Team (JVET) of ITU-T SG, v. 16, p. 19–27, 2019.
- BROSS, B. CE3: Multiple reference line intra prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4). . Em: JVET-L0283. Macao: out. 2018.
- BROSS, B. et al. Versatile Video Coding Editorial Refinements on Draft 10. Disponível em: <https://jvet-experts.org/doc_end_user/current_document.php?id=10540>. Acesso em: 12 fev. 2024.
- BROSS, B. et al. Overview of the Versatile Video Coding (VVC) Standard and its Applications. IEEE Transactions on Circuits and Systems for Video Technology, v. 31, p. 3736–3764, 1 out. 2021.
- CHEN, B. et al. Compact Temporal Trajectory Representation for Talking Face Video Compression. IEEE Transactions on Circuits and Systems for Video Technology, v. 33, n. 11, p. 7009–7023, nov. 2023.
- CHEN, J.; YE, Y.; KIM, S. Algorithm description for versatile video coding and test model 10 (VTM 10). In JVET 19th Meeting, JVET-S2002, Teleconference, jul. 2020.

Comparing ASIC gate-equivalent with XU LUTs. Disponível em: <https://support.xilinx.com/s/question/0D52E00006hpjEmSAI/comparing-asic-gateequivalent-with-xu-luts?language=en_US>. Acesso em: 1 ago. 2024.

DE-LUXÁN-HERNÁNDEZ, S. et al. An Intra Subpartition Coding Mode for VVC. 2019 IEEE International Conference on Image Processing (ICIP). Anais... Em: 2019 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP). set. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8803777>>. Acesso em: 3 ago. 2024

Digital Signal Processing (DSP) Block. Disponível em: <<https://www.intel.com/content/www/us/en/docs/programmable/683152/24-1/digital-signal-processing-dsp-block.html>>. Acesso em: 16 ago. 2024.

GHANBARI, M. Standard Codecs: Image compression to advanced video coding. [s.l.] IET Digital Library, 2003.

GONZALEZ, R. C.; WOODS, R. E. Processamento Digital De Imagens. [s.l.] ADDISON WESLEY BRA, 2009.

GUPTA, P. LUT in FPGA: Lookup Tables (LUT) in FPGA Design (2024). , 1 jul. 2023. Disponível em: <<https://fpgainsights.com/fpga/overview-of-lookup-tables-lut-in-fpga-design/>>. Acesso em: 1 ago. 2024

HUANG, X. et al. Fast algorithms and VLSI architecture design for HEVC intra-mode decision. Journal of Real-Time Image Processing, v. 12, n. 2, p. 285–302, 1 ago. 2016.

JVET - Joint Video Experts Team. Disponível em: <<https://www.itu.int:443/en/ITU-T/studygroups/2017-2020/16/Pages/video/jvet.aspx>>. Acesso em: 12 fev. 2024.

LIN, K. et al. High-Order Intra Prediction for Future Video Coding. 2022 Data Compression Conference (DCC). Anais... Em: 2022 DATA COMPRESSION CONFERENCE (DCC). mar. 2022. Disponível em: <<https://ieeexplore.ieee.org/document/9810756/keywords#keywords>>. Acesso em: 3 ago. 2024

LIU, Z. et al. Deep Multi-Task Learning Based Fast Intra-Mode Decision for Versatile Video Coding. IEEE Transactions on Circuits and Systems for Video Technology, v. 33, n. 10, p. 6101–6116, out. 2023.

PALAU, R. DE C. N. et al. Modern Video Coding: Methods, Challenges and Systems. Journal of Integrated Circuits and Systems, v. 16, n. 2, p. 1–12, 16 ago. 2021.

PAFF, J. et al. Intra Prediction and Mode Coding in VVC. IEEE Transactions on Circuits and Systems for Video Technology, v. 31, n. 10, p. 3834–3847, out. 2021.

POLLACK, J. Displays of A Different Stripe. IEEE Spectrum, v. 43, n. 8, p. 40–44, ago. 2006.

RICHARDSON, I. E. G. H.264 and MPEG-4 video compression: video coding for next generation multimedia. Chichester ; Hoboken, NJ: Wiley, 2003.

RUSSELL. What is a Block RAM in an FPGA? For Beginners. Nandland, 9 jun. 2022. Disponível em: <<https://nandland.com/lesson-15-what-is-a-block-ram-bram/>>. Acesso em: 1 ago. 2024

SALDANHA, M. et al. Performance analysis of VVC intra coding. *Journal of Visual Communication and Image Representation*, v. 79, n. C, 1 ago. 2021.

SALMON, R.; ARMSTRONG, M.; JOLLY, S. Higher Frame rates for more Immersive Video and Television. Disponível em: <<https://www.bbc.co.uk/rd/publications/whitepaper209>>. Acesso em: 16 ago. 2024.

SCHÄFER, M. et al. An Affine-Linear Intra Prediction With Complexity Constraints. 2019 IEEE International Conference on Image Processing (ICIP). Anais... Em: 2019 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP). set. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8803724>>. Acesso em: 3 ago. 2024

SHI, Y.-Q.; SUN, H. Image and Video Compression for Multimedia Engineering : Fundamentals, Algorithms, and Standards, Third Edition. [s.l.] CRC Press, 2019.

STORR, W. D-type Flip Flop Counter or Delay Flip-flop. Basic Electronics Tutorials, 30 ago. 2013. Disponível em: <https://www.electronics-tutorials.ws/sequential/seq_4.html>. Acesso em: 1 ago. 2024

SUEHRING, K. HM-16.20 jvet / HM. Disponível em: <<https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tags/HM-16.20>>. Acesso em: 17 ago. 2024.

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

WIEN, M. High Efficiency Video Coding: Coding Tools and Specification. Berlin, Heidelberg: Springer, 2015.

ZHANG, Q. et al. Fast CU Partition Decision Method Based on Texture Characteristics for H.266/VVC. *IEEE Access*, v. 8, p. 203516–203524, 2020.

ZHANG, Y.; LU, C. Efficient Algorithm Adaptations and Fully Parallel Hardware Architecture of H.265/HEVC Intra Encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 29, n. 11, p. 3415–3429, nov. 2019.