



Thaísa Leal da Silva

CONTRIBUIÇÕES PARA A REDUÇÃO DE COMPLEXIDADE DA CODIFICAÇÃO INTRA NA NORMA HEVC E EXTENSÃO 3D-HEVC

Tese de Doutoramento em Engenharia Eletrotécnica e de Computadores, ramo de especialização em Telecomunicações, orientada pelo Professor Doutor Luís Alberto da Silva Cruz e co-orientada pelo Professor Doutor Luciano Volcan Agostini, apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Fevereiro, 2015



UNIVERSIDADE DE COIMBRA



CONTRIBUIÇÕES PARA A REDUÇÃO DE COMPLEXIDADE DA CODIFICAÇÃO INTRA NA NORMA HEVC E EXTENSÃO 3D-HEVC

Tháísa Leal da Silva

Tese de Doutoramento em Engenharia Electrotécnica e de
Computadores, ramo de especialização em Telecomunicações

Tese submetida ao Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de
Ciências e Tecnologia da Universidade de Coimbra para a obtenção do grau de Doutor.

Orientador:

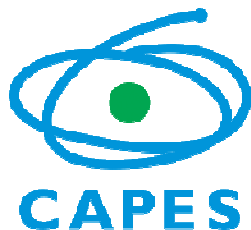
Professor Doutor Luis Alberto da Silva Cruz

Co-orientador:

Professor Doutor Luciano Volcan Agostini

Coimbra, Fevereiro de 2015.

Principal apoio financeiro: CAPES (Ref. 5481/10-4)



AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao meu orientador, Prof. Luis Alberto da Silva Cruz, por todo o seu apoio desde a minha chegada à Portugal, e ao longo desses anos de doutoramento. Agradeço imenso por seus conselhos e críticas, e por sua presença constante sempre pronto a discutir novas ideias e esclarecer minhas dúvidas. Sua orientação e o conhecimento que me transmitiu foram essenciais para a realização desse trabalho.

Agradeço também ao meu co-orientador, Prof. Luciano Volcan Agostini, pelo apoio que me deu, mesmo a distância, por estar sempre disposto a conversar e me incentivar a seguir em frente.

Gostaria de agradecer, em especial, à minha família, a quem dedico esta tese, por todo o carinho e força que sempre me deram ao longo desta jornada. Aos meus pais, Orlando e Eva, por tudo que me ensinaram e ainda ensinam, e por confiarem nas minhas decisões. Seu amor me fez mais forte mesmo quando a saudade era enorme, e a fé que me transmitiram, me fez seguir adiante mesmo nos momentos mais difíceis.

À minha irmã, Deise, por todas as palavras de afeto, e por cuidar dos nossos pais nesse tempo em que estive ausente. Ao meu cunhado Daniel pela grande amizade, e por ser um pai tão amável e presente para o meu afilhado Davi. Ao Davi, que nasceu logo que vim para Portugal, e que agora já está um rapazinho, pela compreensão de ter uma madrinha tão distante e por me transmitir muito amor assim mesmo.

Ao Lauro, por seu companheirismo e sua presença sempre otimista, buscando me mostrar o lado bom das coisas e fortalecendo minha crença de que podemos fazer o mundo melhor. Por sua grande amizade e amor, e pelo apoio que me deu durante esses anos de doutoramento.

Gostaria ainda de agradecer ao Instituto de Telecomunicações (IT), à Fundação para Ciência e Tecnologia (FCT), e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) do Brasil, pelo suporte financeiro que me concederam, e que tornou possível o desenvolvimento deste trabalho de doutoramento. Muito obrigada.

RESUMO

A popularização de dispositivos eletrônicos de consumo que suportam vídeos de alta definição, tais como *smartphones* e câmeras digitais, constitui um obstáculo para os padrões de compressão de vídeo atuais, os quais ainda não estão aptos a suportar altas taxas de compressão, com baixa perda de qualidade e complexidade computacional aceitável. Visando resolver este problema, o grupo JCT-VC desenvolveu um novo padrão de compressão de vídeo, chamado HEVC, o qual alcança uma redução na taxa de bits consideravelmente maior quando comparado aos seus predecessores. Um dos fatores responsáveis por esse ganho em eficiência de codificação é o novo método de codificação intra adotado pelo HEVC, o qual introduz um grande número de direções de predição, melhorando o desempenho da taxa-distorção, a custo de um aumento considerável na complexidade computacional.

Além disso, com os recentes avanços em tecnologia de vídeo tridimensional (3D), uma considerável expansão de sistemas de vídeo 2D para sistemas 3D tem ocorrido na indústria de multimídia e, conseqüentemente, a demanda por armazenamento e transmissão de conteúdo de vídeo 3D tem crescido muito. Para atender essa demanda e tirar partido do ganho em eficiência alcançado pelo padrão HEVC, o grupo JCT-3V iniciou o desenvolvimento de uma extensão 3D, chamada 3D-HEVC, a qual se encontra em fase final de padronização.

Esta tese apresenta soluções algorítmicas para a redução de complexidade da codificação intra nos cenários de vista única (*Singleview*) e multivista (*Multiview*), contribuindo para reduzir de forma significativa a complexidade global do codificador HEVC e sua extensão 3D-HEVC. Foram desenvolvidos algoritmos rápidos para a decisão de modo intra do HEVC, decisão da profundidade das árvores de codificação inter-vistas, predição intra com inferência inter-vistas, codificação intra dos mapas de profundidade, e codificação intra de textura mais profundidade no contexto da extensão 3D-HEVC.

Os resultados experimentais demonstraram que as soluções propostas para a codificação intra do HEVC, chegaram a alcançar uma redução de complexidade

computacional de aproximadamente 40% em média, com perdas insignificantes em termos de eficiência de codificação (1,7% em *BD-Rate* em média) quando comparadas à codificação intra do HEVC. Além disso, a partir das soluções propostas para a codificação intra das vistas de textura e mapas de profundidade na extensão 3D-HEVC, foi possível alcançar uma redução de 26%, em média, na complexidade global do codificador 3D-HEVC, com uma perda de desempenho de 0,32% em *BD-Rate* em média, para as vistas originais do vídeo, e 0,45% em *BD-Rate* em média, para as vistas sintetizadas.

Palavras-chave: codificação de vídeo, HEVC, predição intra, análise de textura, 3D-HEVC, predição inter-vista, redução de complexidade.

ABSTRACT

The popularization of consumer electronic devices supporting high-definition videos, like smart phones, digital cameras, etc., is a challenge to the current standard video coders which do not achieve high compression rates with low quality losses at acceptable complexity levels. In order to find better encoding methods for high-definition contents, the JCT-VC developed a new video coding standard, called HEVC, which achieves a higher encoding efficiency when compared to its predecessors. One of the factors responsible for this improvement in encoding efficiency is the new intra coding method adopted by the HEVC, which introduces a larger number of prediction directions, enhancing the rate-distortion (RD) performance at the cost of a higher computational complexity.

Moreover, with the recent improvements in three dimensional (3D) video technology, a considerable expansion from 2D to 3D systems has occurred in the multimedia industry and consequently, the demand for storage and transmission of the 3D video content has grown. To cope with this demand and take advantage of the enhanced coding efficiency achieved by the HEVC standard, the JCT-3V started the development of a 3D extension to the HEVC, called 3D-HEVC, which is near the final standardization stage.

This thesis presents algorithmic solutions to reduce the intra coding complexity in the Singleview and Multiview scenarios, contributing to reduce significantly the overall complexity of the HEVC encoder and its 3D-HEVC extension. Fast algorithms have been developed for the HEVC intra mode decision, inter-view prediction of coding tree decision depth, intra coding with inter-view mode prediction, intra coding of depth maps, and intra coding of texture and depth of the 3D-HEVC.

The experimental results demonstrated that the proposed HEVC intra coding solutions achieved a computational complexity reduction of approximately 40% on average, with negligible losses in terms of coding efficiency (1.7% in BD-Rate on average) when compared to the HEVC intra coding. Furthermore, the proposed solutions for the

intra coding of 3D-HEVC texture views and depth maps achieved an average reduction of 26% for the overall complexity of the 3D-HEVC encoder with minimal loss in performance (0.32% on average in BD-Rate) for the video only, and 0.45% BD-Rate increase, on average, for the synthesized views.

Keywords: video coding, HEVC, intra prediction, texture analysis, 3D-HEVC, inter-view prediction, complexity reduction.

CONTEÚDO

AGRADECIMENTOS.....	i
RESUMO	iii
ABSTRACT.....	v
CONTEÚDO.....	vii
LISTA DE FIGURAS	xi
LISTA DE TABELAS	xv
LISTA DE ABREVIATURAS	xvii
1. INTRODUÇÃO	1
1.1. Contexto e Motivação.....	1
1.2. Objetivos do Trabalho	4
1.3. Contribuições da Tese.....	5
1.3.1. Redução de Complexidade da Predição Intra-Tramas do HEVC e HEVC Multivista.	6
1.3.2. Algoritmos Rápidos para a Decisão de Profundidade da Árvore de Codificação	7
1.3.3. Redução da Complexidade de Codificação Intra das Vistas de Textura e Profundidade do 3D-HEVC.....	7
1.4. Organização da Tese.....	8
2. CODIFICAÇÃO DE VÍDEO.....	11
2.1. Conceitos Básicos.....	11
2.2. Técnicas de Compressão de Dados de Vídeo	12
2.3. Critérios de Avaliação	14
2.4. O Padrão HEVC.....	17
2.4.1. Estrutura de Codificação.....	18
2.4.1.1. Unidade de Codificação (CU)	18
2.4.1.2. Unidade de Predição (PU).....	19
2.4.1.3. Unidade de Transformação (TU)	20

2.4.1.4.	Estrutura de <i>Slices e Tiles</i>	20
2.4.2.	Ferramentas de Codificação	22
2.4.2.1.	Predição Intra-Trama.....	23
2.4.2.2.	Predição Inter-Trama.....	25
2.4.2.1.	Transformada e Quantização.....	27
2.4.2.2.	Codificação de Entropia	28
2.4.2.3.	Malha de Filtragem	28
2.4.2.3.1.	Filtro de Redução de Efeito de Bloco	28
2.4.2.3.2.	Deslocamento Adaptativo à Amostra (SAO)	29
2.4.3.	Estruturas de Predição Temporal.....	30
2.4.3.1.	<i>Intra-only</i> ou <i>All-Intra</i> (AI).....	30
2.4.3.2.	<i>Low-delay</i> (LD)	31
2.4.3.3.	<i>Random-Access</i> (RA)	31
3.	CODIFICAÇÃO DE VÍDEO EM 3D.....	33
3.1.	Sistemas de Vídeo 3D.....	33
3.2.	A extensão 3D-HEVC	34
3.2.1.	Formato de Dados e Estrutura do Sistema.....	34
3.2.2.	Processo de Codificação	36
3.2.3.	Ferramentas de Codificação	37
3.2.3.1.	Predição com Compensação de Disparidade.....	38
3.2.3.2.	Predição Inter-Vistas de Movimento e Resíduo	38
3.2.3.3.	Compensação de Iluminação (IC)	39
3.2.4.	Codificação de Mapas de Profundidade	41
3.2.4.1.	Modos de Modelagem da Profundidade (DMMs).....	41
3.2.4.1.1.	Sinalização Explícita da Wedgelet.....	43
3.2.4.1.2.	Predição Inter-componente das Partições de Contorno.....	45
3.2.4.2.	Herança dos Parâmetros de Movimento.....	45
3.2.4.3.	Predição das Quadrees de Profundidade	46
4.	ALGORITMOS RÁPIDOS PARA A CODIFICAÇÃO INTRA DO HEVC.....	49
4.1.	Otimização da Predição Intra.....	50
4.1.1.	Orientação da Borda Dominante	56
4.1.2.	Correlação entre as PUs de Níveis de Profundidade Adjacentes.....	67
4.1.3.	Verificação de Modo-Fronteira	71
4.2.	Análise de Resultados e Comparações	75

5.	ALGORITMOS RÁPIDOS PARA CODIFICAÇÃO INTRA DA EXTENSÃO 3D-HEVC...	79
5.1.	HEVC <i>Simulcast</i> vs. HEVC Multivista	79
5.1.1.	Predição Inter-Vista do Modo Intra.....	82
5.1.2.	Predição Inter-Vistas da Profundidade da Árvore de Codificação	91
5.2.	Algoritmos de Aceleração para o Codificador 3D-HEVC.....	97
5.2.1.	Otimização da Decisão de Profundidade Máxima da Árvore de CUs.....	97
5.2.2.	Redução de Complexidade da Codificação Intra dos Mapas de Profundidade	102
5.2.3.	Redução de Complexidade da Codificação Intra de Textura e Profundidade	109
5.2.3.1.	Determinação de Borda	110
5.2.3.2.	Seleção dos Modos Candidatos	112
5.3.	Análise de Resultados e Comparações	118
6.	CONCLUSÕES E TRABALHOS FUTUROS	123
7.	REFERÊNCIAS	127
	APÊNDICE A: Lista de Artigos Científicos Publicados e Comunicações Proferidas.....	137
	APÊNDICE B: Cenário das Simulações - Condições Comuns de Teste	141

LISTA DE FIGURAS

Figura 1.1: Exemplo de vista sintetizada a partir da textura e profundidade de vistas de câmeras adjacentes do vídeo PoznanStreet.	3
Figura 1.2: Exemplo de mapa de profundidade do vídeo UndoDancer, destacando as áreas homogêneas e bordas bem definidas.	4
Figura 2.1: Exemplo de formatos de subamostragem de cores [33].	12
Figura 2.2: Exemplo de curvas de Bjøntegaard-Delta [41].	16
Figura 2.3: Exemplo de estrutura de codificação em árvore quadrática.	19
Figura 2.4: Estrutura de Codificação: Unidades de Codificação (CU), Predição (PU) e Transformação (TU).	20
Figura 2.5: Exemplo de estrutura de codificação de segmentos de <i>slice</i> independentes e dependentes [46].	21
Figura 2.6: Exemplo de particionamento de uma trama em <i>Slices</i> (a) e em <i>Tiles</i> (b) [5].	22
Figura 2.7: Diagrama em blocos de um codificador HEVC.	22
Figura 2.8: Direções de predição intra no HEVC [49].	23
Figura 2.9: Predição planar de uma PU 8x8, amostras interpoladas linearmente (a) e bi-linearmente (b) [49].	24
Figura 2.10: Mapeamento entre a direção e o modo de predição intra [44].	24
Figura 2.11: Posições das PUs espacialmente candidatas para o Modo <i>Merge</i> [44].	26
Figura 2.12: Antes da aplicação do <i>Deblocking Filter</i> (a) e após a aplicação do <i>Deblocking Filter</i> (b).	29
Figura 2.13: Sequência de vídeo codificada utilizando configuração <i>Intra-only</i> [44].	30
Figura 2.14: Sequência de vídeo codificada utilizando configuração <i>Low-Delay</i> [44].	31
Figura 2.15: Sequência codificada utilizando configuração <i>Random-Access</i> [44].	32
Figura 3.1: Representação do formato vídeo MVD com tramas de textura e mapas de profundidade associados [16].	35
Figura 3.2: Estrutura das Unidades de Acesso e Ordem de Codificação dos Componentes de Textura e Profundidade [11][44].	36
Figura 3.3: Estrutura básica do codificador 3D-HEVC [11].	37
Figura 3.4: Exemplo de DCP e MCP para codificação de vistas dependentes [11].	38
Figura 3.5: Vetor de movimento predito a partir do bloco correspondente da vista de referência já codificada [12][44].	39
Figura 3.6: Variação de iluminação e cromaticidade em tramas capturadas por câmeras diferentes [63].	40
Figura 3.7: Amostras vizinhas usadas no processo de compensação de iluminação [11].	40

Figura 3.8: Exemplo de mapa de profundidade da sequência PoznanStreet, destacando as características de bordas bem definidas e áreas homogêneas.	42
Figura 3.9: Partição Wedgelet (a) e Partição de Contorno (b): Espaço de sinal contínuo (à esquerda) e padrão de partição correspondente (à direita). Trama de profundidade da sequência UndoDancer (c) exemplo de informação de Partição Wedgelet (azul) e Partição de Contorno (verde) [61].....	43
Figura 3.10: Seleção das Partições Wedgelet [11].	44
Figura 3.11: Partições de profundidade possíveis a partir das partições de textura utilizadas [11].	47
Figura 4.1: Tipos de particionamento de PUs intra.	50
Figura 4.2: Direções de Predição Intra dos padrões H.264/AVC (a) e HEVC (b).	51
Figura 4.3: Distribuição da complexidade de codificação usando <i>all-intra</i> no codificador HM. ...	51
Figura 4.4: Decisão de modo na predição intra do HEVC.	52
Figura 4.5: Análise de complexidade dos três principais módulos da predição intra do HM [76].	54
Figura 4.6: Cálculo de borda para uma PU 16x16.	57
Figura 4.7: Conjuntos de orientação de borda.	58
Figura 4.8: Redução de Complexidade para cada tamanho de subconjunto avaliado.	59
Figura 4.9: BD-Rate para cada tamanho de subconjunto avaliado.	59
Figura 4.10: Conjunto de modos vertical versus frequência de utilização normalizada.	60
Figura 4.11: Conjunto de modos horizontal versus frequência de utilização normalizada.	61
Figura 4.12: Conjunto de modos diagonal 45° versus frequência de utilização normalizada.	61
Figura 4.13: Conjunto de modos diagonal 135° versus frequência de utilização normalizada.	61
Figura 4.14: Conjunto de modos intra versus frequência de utilização normalizada.	62
Figura 4.15: Subconjuntos de modos para cada orientação de borda.	63
Figura 4.16 : Exemplo de tipos de bordas dominantes de acordo com a textura da PU.	64
Figura 4.17: Fluxograma do algoritmo proposto para a decisão de modo intra baseado na orientação de borda dominante.	64
Figura 4.18: Estrutura de árvore hierárquica das PUs no HEVC.	68
Figura 4.19: Fluxograma do algoritmo proposto para a decisão de modo intra baseado na orientação de borda e na estrutura de árvore hierárquica do HEVC.	70
Figura 4.20: Subconjunto de modos para cada orientação de borda incluindo a representação do teste de modo-fronteira.	72
Figura 4.21: Fluxograma do algoritmo proposto para a decisão de modo intra utilizando a verificação de modo-fronteira.	73
Figura 5.1: Exemplo da estrutura de predição do HEVC Multivista.	80
Figura 5.2: Codificação <i>all-intra</i> usando o esquema proposto para codificação HEVC baseada em múltiplas vistas.	84
Figura 5.3: Predição inter-vista do modo intra proposto para a extensão multivista do HEVC.	84
Figura 5.4 – Fluxograma do método de predição inter-vista do modo intra proposto para a codificação de vídeo multivista baseada no codificador HEVC.	85

Figura 5.5: Exemplo de estrutura de codificação usando quadrees.	92
Figura 5.6: Predição inter-vista da profundidade da árvore de codificação para a extensão multivista do HEVC.	93
Figura 5.7: Fluxograma do método de predição inter-vistas da profundidade da árvore de codificação.	94
Figura 5.8: Fluxograma do algoritmo proposto para otimização inter-vistas das decisões de profundidade máxima da árvore de CUs.	99
Figura 5.9: Algoritmo proposto para a predição inter-vistas das decisões de profundidade da árvore de CUs do 3D-HEVC.	100
Figura 5.10: Processo de decisão de modo intra do 3D-HEVC.	102
Figura 5.11: Fluxograma do algoritmo rápido para a decisão de modo intra dos mapas de profundidade.	105
Figura 5.12: Cálculo de borda de uma PU 8x8.	106
Figura 5.13: Subconjuntos de modos para cada uma das cinco orientações de borda.	107
Figura 5.14: Máscaras do operador de Sobel.	110
Figura 5.15: Exemplo de aplicação das máscaras Sobel utilizando uma trama do vídeo Balloons: Imagem original (a), Máscara de Sobel Horizontal (b), Máscara de Sobel Vertical (c), Máscaras de Sobel Combinadas (d).	111
Figura 5.16: Modos angulares e direção ϕ associada.	112
Figura 5.17: Direção de gradiente e direção de borda.	113
Figura 5.18: Exemplo de um histograma de direções de modo para uma PU 32x32.	114
Figura 5.19: Curvas RD do algoritmo proposto em relação ao HTM 12.1 (GTFly 1920x1080 pixels).	116
Figura 5.20: Comparação do tempo de codificação do algoritmo proposto em relação ao HTM12.1 (GTFly 1920x1080 pixels).	117
Figura 5.21: Curvas RD do algoritmo proposto em relação ao HTM 12.1 (Newspaper 1024x768 pixels).	117
Figura 5.22: Comparação do tempo de codificação do algoritmo proposto em relação ao HTM12.1 (Newspaper 1024x768 pixels).	118
Figura B.1: Trama de textura (a) e Trama de profundidade (b) da sequência Balloons.	142
Figura B.2: Trama de textura (a) e Trama de profundidade (b) da sequência Newspaper.	143
Figura B.3: Trama de textura (a) e Trama de profundidade (b) da sequência Kendo.	143
Figura B.4: Trama de textura da sequência GTFly.	144
Figura B.5: Trama de profundidade da sequência GTFly.	144
Figura B.6: Trama de textura da sequência PoznanStreet.	145
Figura B.7: Trama de profundidade da sequência PoznanStreet.	145
Figura B.8: Trama de textura da sequência UndoDancer.	146
Figura B.9: Trama de profundidade da sequência UndoDancer.	146
Figura B.10: Trama de textura da sequência PoznanHall2.	147
Figura B.11: Trama de profundidade da sequência PoznanHall2.	147

Figura B.12: Trama de textura da sequência Shark.....	148
Figura B.13: Trama de profundidade da sequência Shark.	148
Figura B.14: Trama de textura da sequência PoznanCarPark.	149
Figura B.15: Trama de profundidade da sequência PoznanCarPark.....	149
Figura B.16: Trama de textura da sequência PoznanHall1.	150
Figura B.17: Trama de profundidade da sequência PoznanHall1.	150

LISTA DE TABELAS

Tabela 2.1: Coeficientes do filtro de 7-taps e 8-taps para interpolação de posições de 1/4 e 2/4 de pixel para a componente de luminância [44].....	27
Tabela 2.2: Coeficientes do filtro de 4-taps para interpolação de posições de 1/8 de pixel para as componentes de crominância [44].....	27
Tabela 4.1: Características das sequências de vídeo.	65
Tabela 4.2: Comparação do algoritmo de Orientação de borda dominante proposto em relação ao algoritmo usado no HM10.....	66
Tabela 4.3: Correlação na orientação de borda das PUs inter-níveis.	68
Tabela 4.4: Comparação do algoritmo de Correlação entre a Orientação de Borda das PUs Inter-níveis proposto em relação ao algoritmo usado no HM10.	71
Tabela 4.5: Comparação do algoritmo de Verificação do modo-fronteira proposto em relação ao algoritmo usado no HM10.....	74
Tabela 4.6: Comparação da redução de complexidade do algoritmo proposto em relação aos trabalhos relacionados.....	77
Tabela 5.1: Comparação entre HEVC Simulcast e HEVC Multivista.	81
Tabela 5.2: Modos intra da vista ímpar em relação aos modos intra das vistas pares.	83
Tabela 5.3: Comparação do método de predição inter-vista do modo intra em relação ao HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 2.....	86
Tabela 5.4: Sequências de teste para experimentos adicionais.	87
Tabela 5.5: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 2.....	88
Tabela 5.6: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 3.....	89
Tabela 5.7: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 4.....	90
Tabela 5.8: Comparação do método de predição inter-vistas proposto em relação ao HEVC Simulcast considerando a codificação da Vista 1.	95
Tabela 5.9: Comparação do método de predição inter-vistas proposto em relação ao HEVC Simulcast considerando a codificação da Vista 2.	96
Tabela 5.10: Análise de co-ocorrências da profundidade máxima das CUs da Vista 2 em relação a profundidade máxima das CUs das vistas 0 e 1.	98
Tabela 5.11: Comparação do método de predição inter-vistas proposto em relação ao 3D-HEVC considerando a codificação da Vista 2.	101
Tabela 5.12: Modos de predição intra do 3D-HEVC.	103
Tabela 5.13: Porcentagem de ocorrências dos modos intra no 3D-HEVC.	104

Tabela 5.14: Correlação entre os MPMs e a escolha do modo Planar como melhor modo intra...	104
Tabela 5.15: Resultados de BD-rate e tempo de processamento para 3 vistas considerando o método proposto para a codificação intra dos mapas de profundidade, caso <i>all-Intra</i>	108
Tabela 5.16: Características das sequências de teste.....	115
Tabela 5.17: Resultados de BD-rate e tempo de processamento para 3 vistas considerando o método proposto para a codificação intra das vistas de textura e profundidade, caso <i>all-Intra</i>	115
Tabela B.1: Características das sequências de teste das CTCs.	141

LISTA DE ABREVIATURAS

2D	Two-Dimensional
3D	Three-Dimensional
3D-HEVC	3D High Efficiency Video Coding
AC	Arithmetic Coding
AVC	Advanced Video Coding
B	Bi-predictive
CABAC	Context-Based Adaptive Binary Arithmetic Coding
CAVLC	Context-Based Adaptive Variable Length Coding
Cb	Chrominance blue
codec	codificador/descodificador
Cr	Chrominance red
CTC	Common Test Conditions
CTU	Coding Tree Unit
CU	Coding Unit
DC	Direct Current
DCP	Disparity-Compensated Prediction
DCT	Discrete Cosine Transform
DIBR	Depth Image Based Rendering
DMM	Depth Modelling Modes
GPB	Generalized P and B
GOP	Group Of Pictures
FPS	Frames per second
FTV	Free viewpoint Television
FVV	Free viewpoint Video
FullHD	Full High Definition
HD	High Definition

HEVC	High Efficiency Video Coding
HM	HEVC Test Model
HTM	3D-HEVC Test Model
IEC	International Electrotechnical Commission
IEEE	Institute of Electric and Electronics Engineers
ISO	International Organization for Standardisation
ITU-T	International Telecommunication Union - Telecommunication
JCT-3V	Joint Collaborative Team on 3D Video Coding Extension Development
JCT-VC	Joint Collaborative Team on Video Coding
JVT	Joint Video Team
MC	Motion Compensation
MCP	Motion Compensated Prediction
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MPI	Motion Parameter Inheritance
MPM	Most Probable Mode
MSE	Mean Square Error
MVD	Multiview plus Depth
NAL	Network Abstraction Layer
P	Predictive
PDA	Personal Digital Assistant
POC	Picture Order Count
PSNR	Peak Signal-to-Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter
RD	Rate Distortion
RDOQ	Rate-Distortion Optimized Quantization
RDO	Rate Distortion Optimization
RGB	Red, Green, Blue
RQT	Residual Quadtree

SAD	Sum of Absolute Differences
SATD	Hadamard transformed SAD
SAO	Sample Adaptive Offset
TU	Transform Unit
TV	Televisão
UHD	Ultra High Definition
VCEG	Video Coding Experts Group
VLC	Variable Length Coding
VSRS	View Synthesis Reference Software
Y	Luminance
YCbCr	Luminance, Chrominance Blue, Chrominance Red

1. INTRODUÇÃO

Neste capítulo, será apresentada uma introdução ao trabalho de pesquisa desenvolvido e descrito no decorrer dessa tese. Inicialmente, o escopo e a motivação do trabalho são apresentados, bem como a descrição dos objetivos. A seguir, as principais contribuições do trabalho são listadas, seguidas por um resumo do conteúdo que será apresentado em detalhe nos capítulos seguintes da tese.

1.1. Contexto e Motivação

Os recentes avanços em dispositivos de captura e exibição de vídeo e a crescente popularização de vídeos com alta e ultra alta definição (HD e UHD) e conteúdos de vídeo 3D (*Three-Dimensional*) em aplicações no mercado de multimídia (ex.: televisões e *Blu-rays* 3D, transmissões em FTV e sistemas de videovigilância) introduziram novos desafios para a tecnologia de compressão de vídeo. Deste modo, a qualidade dos codificadores de imagens e vídeos é essencial para o sucesso de produtos que manipulam este tipo de mídia.

Em termos de codificadores de vídeo, o padrão H.264/AVC (*Advanced Video Coding*) [1][2][3] de compressão de vídeo, consegue ganhos significativos, quando comparado com os padrões que o antecederam, como o MPEG-2 e o MPEG-4, conseguindo reduzir o fluxo de dados necessário para comprimir os vídeos, chegando a duplicar as taxas de compressão dos padrões anteriores.

Contudo, o codificador H.264/AVC ainda apresenta taxas de bits muito elevadas o que afeta seu desempenho, tornando imperativo o desenvolvimento de novas soluções algorítmicas para compressão de vídeos de alta definição. Em função disso, um novo padrão de compressão de vídeo, chamado de HEVC (*High Efficiency Video Coding – Codificação de Vídeo de Alta Eficiência*) foi recentemente desenvolvido [4][5], visando alcançar uma eficiência de codificação maior que a do H.264/AVC, com um nível de complexidade não superior à do perfil *Baseline* do H.264/AVC [6].

Para viabilizar um melhor compromisso eficiência-complexidade, diversas ferramentas de codificação de vídeo do H.264/AVC foram aperfeiçoadas e novas ferramentas introduzidas, entre elas: estrutura de codificação em árvore (*quadtree*), particionamento de bloco flexível, maior número de direções de predição e tamanhos de blocos maiores, permitindo ao codificador prover uma redução na taxa de bits de aproximadamente 50% em comparação ao H.264/AVC com o mesmo nível de qualidade visual perceptiva [5]. No entanto, este aumento da eficiência foi alcançado a custo de um grande acréscimo na complexidade computacional do codificador.

Além disso, considerando a codificação de vídeos com múltiplas vistas, atualmente a extensão multivista do padrão H.264/AVC, chamada MVC (*Multiview Video Coding*) [7] é o estado-da-arte em codificação multivista. No entanto, a redução da taxa de bits quando comparada à codificação das vistas de forma independente utilizando o H.264/AVC é de apenas cerca de 10-30%, a qual não é suficiente para suprir as exigências da indústria de comunicações, especialmente para as novas aplicações de transmissão de vídeo 3D [8][9][10]. Para atender essa necessidade e suportar a crescente demanda por armazenamento e transmissão de conteúdo de vídeo 3D, a JCT-3V (*Joint Collaborative Team on 3D Video Coding Extension Development*) iniciou o desenvolvimento de uma extensão multivista para o padrão HEVC, denominada 3D-HEVC (*3D High Efficiency Video Coding*) [11][12][13]. O padrão emergente 3D-HEVC inclui todas as funcionalidades adotadas pelo HEVC para codificação de vídeo 2D (*Two-Dimensional*) e, além disso, possui suporte à codificação de vídeo 3D representado usando o formato MVD (*Multiview plus Depth*) de vídeo multivista mais profundidade [12][14].

No formato MVD, cada vista do sinal de vídeo é constituída por uma componente de textura e outra de profundidade e ambas são codificadas e multiplexadas em um *bitstream* 3D. Depois de receber a informação de textura e profundidade, o decodificador é capaz de gerar vistas virtuais intermédias (vistas sintetizadas) espacialmente posicionadas entre as vistas transmitidas (Figura 1.1), utilizando a técnica de renderização baseada na imagem de profundidade (*Depth Image Based Rendering* - DIBR) [12][15].



Figura 1.1: Exemplo de vista sintetizada a partir da textura e profundidade de vistas de câmeras adjacentes do vídeo PoznanStreet.

O formato MVD permite que vistas intermédias virtuais possam ser geradas a partir de qualquer posição entre as câmeras, proporcionando uma funcionalidade FVV (*free viewpoint video*) avançada, com livre navegação entre as câmeras originais, e permitindo uma considerável redução no processamento de dados do vídeo no que diz respeito ao armazenamento e largura de banda de transmissão, devido ao número reduzido de vistas a serem transmitidas [16].

A componente de profundidade de uma trama (ou *frame*) de um vídeo 3D, também chamada de mapa de profundidade, é semelhante a uma imagem monocromática, onde os diferentes tons de cinzento representam os valores de distância entre a câmera e os objetos da cena. Os mapas de profundidade são caracterizados por grandes áreas homogêneas e bordas bem definidas (Figura 1.2), as quais geralmente coincidem com os limites dos objetos da cena.

Com relação à compressão dos mapas de profundidade, dado que os modos intra convencionais (adotados no HEVC), quando utilizados para codificar os mapas de profundidade, podem gerar artefatos perceptíveis próximos às bordas dos objetos das vistas sintetizadas, novos modos intra chamados DMMs (*Depth Modelling Modes*) [11] foram adicionados à predição intra do 3D-HEVC buscando melhorar a codificação da informação de profundidade nessas bordas.

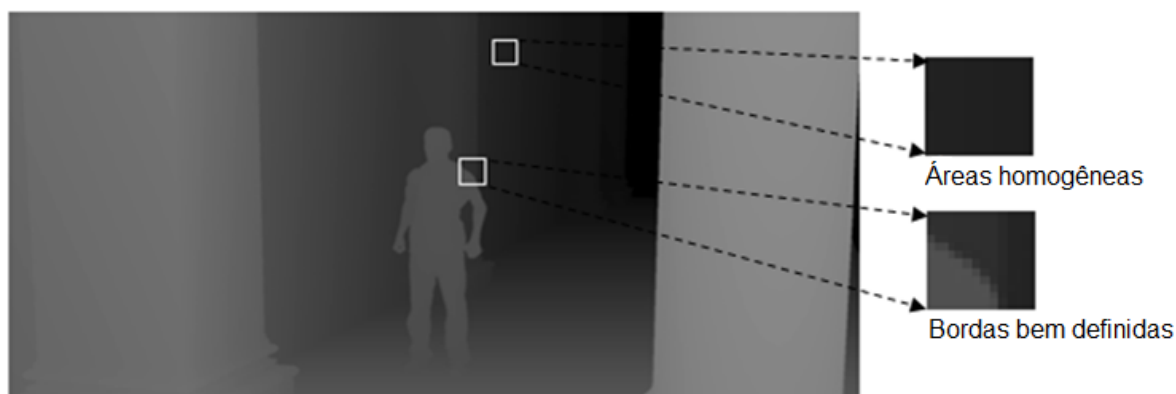


Figura 1.2: Exemplo de mapa de profundidade do vídeo UndoDancer, destacando as áreas homogêneas e bordas bem definidas.

Assim, tendo em vista as novas características e os desafios introduzidos pelo padrão HEVC e pela sua extensão multivista 3D-HEVC, esta tese apresenta o desenvolvimento de novas heurísticas de aceleração de algoritmos para codificação intra, em ambos os cenários: vista única (*Singleview*) e múltiplas vistas (*Multiview*).

Para tanto, foram investigadas algumas das principais ferramentas de codificação adotadas pelo padrão HEVC e, posteriormente, a sua aplicação no codificador multivista. Entre essas ferramentas, a codificação intra-tramas e a codificação em árvore das CUs (*Coding Units*) [17][18] foram exploradas e otimizadas, especialmente pelo significativo aumento que elas acarretam à complexidade computacional do padrão HEVC [19][20][21].

Além disso, dado que o padrão 3D-HEVC ainda está em desenvolvimento, novas contribuições algorítmicas ainda podem ser atribuídas ao padrão de modo a melhorar o compromisso entre a performance de codificação e tempo de processamento. Assim, considerando os modos adicionados à predição intra do 3D-HEVC e a complexidade acrescida ao codificador associada à codificação dos mapas de profundidade, algoritmos rápidos para decisão de modo foram desenvolvidos, otimizando o processo de codificação intra e reduzindo o tempo de processamento do 3D-HEVC.

1.2. Objetivos do Trabalho

O objetivo geral deste trabalho é compreender o comportamento do codificador HEVC e do codificador 3D-HEVC, e propor algoritmos capazes de atender as exigências do

mercado de multimídia, no que diz respeito à complexidade de codificação de vídeos de alta resolução e de sistemas de vídeo em 3D. Nesse contexto, algumas das principais ferramentas adotadas pelo padrão HEVC e pela sua extensão multivista, o 3D-HEVC, foram cuidadosamente estudadas e novas soluções de aceleração dos algoritmos de codificação foram desenvolvidas, as quais serão apresentadas no decorrer desta tese.

Assim, de forma resumida, os objetivos específicos deste trabalho estão descritos a seguir:

- Estudo do estado da arte em técnicas de codificação de vídeo com foco no padrão HEVC;
- Desenvolvimento de soluções visando otimizar o processo de predição intra-tramas utilizado pelo padrão HEVC, com o intuito de alcançar um melhor compromisso entre o desempenho e a complexidade da codificação;
- Desenvolvimento de novas soluções para simplificar a decisão da profundidade máxima da árvore de codificação, utilizando o padrão HEVC em modo *Simulcast* (comumente chamado de *HEVC Simulcast*) e, posteriormente, a extensão multivista MV-HEVC (*Multiview High Efficiency Video Coding*).
- Estudo do estado da arte em sistemas de codificação multivista e análise das técnicas adotadas na extensão 3D-HEVC, buscando acelerar a codificação intra-tramas aplicada aos mapas de profundidade;
- Utilização de métodos de análise de textura, visando a redução da complexidade na codificação das vistas de textura e profundidade do padrão 3D-HEVC, com ênfase na otimização da decisão de modo intra.

1.3. Contribuições da Tese

Nesta seção, será apresentado um resumo das principais contribuições resultantes do trabalho de pesquisa e desenvolvimento envolvidos nesta tese. Tais contribuições estão relacionadas, principalmente, com inovações algorítmicas desenvolvidas visando a redução de complexidade do processo de Predição Intra, tanto no âmbito da codificação *Singleview* utilizando o padrão HEVC, como da codificação *Multiview* utilizando inicialmente o *HEVC*

Simulcast e, logo após, a extensão 3D-HEVC. Como resultado dessas inovações, obteve-se uma redução efetiva na complexidade global desses codificadores.

Uma descrição aprofundada das contribuições desta tese, incluindo detalhes técnicos de implementação será apresentada nos Capítulos 4 e 5.

Tais contribuições estão diretamente relacionadas com os objetivos do trabalho descritos na seção 1.2, conforme apresentado a seguir:

1.3.1. Redução de Complexidade da Predição Intra-Tramas do HEVC e HEVC Multivista

Uma das principais inovações do padrão HEVC, é o significativo aumento no conjunto de modos de predição direcional intra quando comparado aos padrões anteriores.

A predição intra do HEVC contém um total de 35 modos de predição, a partir dos quais o melhor modo em termos de taxa de distorção é escolhido através de um método de pesquisa bastante complexo, consumindo um tempo de processamento significativo. Uma vez que a elevada complexidade da predição intra é uma importante questão de pesquisa, este trabalho focou-se na busca por soluções que permitissem a redução dessa complexidade, sem afetar de forma significativa a eficiência da codificação.

A redução na complexidade da predição intra é especialmente importante para aplicações que têm por objetivo o processamento de vídeo em tempo real em dispositivos com restrições computacionais e/ou de potência, tais como equipamentos de videovigilância e aparelhos portáteis de baixo custo.

Este trabalho apresenta novos e eficientes métodos para otimização da decisão de modo da predição intra, reduzindo o tempo de processamento da codificação intra e, conseqüentemente, a complexidade total do codificador.

Esses métodos foram desenvolvidos principalmente através da exploração da textura das PUs para restringir o número de modos direcionais a serem avaliados na decisão de modo intra e, da correlação existente entre as PUs de níveis de profundidade adjacentes na árvore de codificação do HEVC. Tais métodos e otimizações dos mesmos foram publicados em [22]-[28].

Além disso, a partir dos métodos rápidos para a predição *Singleview*, um novo método de predição intra com inferência de modos inter-vista foi também desenvolvido, o

qual explora a correlação entre as direções de modo intra pertencentes a vistas vizinhas no vídeo, com o objetivo de acelerar o processo de predição intra em aplicações de codificação multivista. Para realizar essa exploração foram utilizados o *HEVC Simulcast* e o MV-HEVC [11]. Tal método de predição intra inter-vistas foi publicado em [29].

1.3.2. Algoritmos Rápidos para a Decisão de Profundidade da Árvore de Codificação

O novo padrão HEVC alcança melhor desempenho de compressão em comparação com os padrões anteriores, com igual qualidade de vídeo perceptual, enquanto cobre uma ampla gama de aplicações. No entanto, para suportar aplicações mais avançadas, que incluem representações multivista e estereoscópica, uma extensão multivista do padrão HEVC está em desenvolvimento com previsão de ser concluída em breve.

Esta extensão irá proporcionar grandes melhorias no desempenho da codificação de vídeo com múltiplas vistas, mas a custo de uma elevada complexidade computacional.

Tendo em vista que a profundidade da árvore de codificação também tem influência na complexidade computacional da codificação intra, a qual é o foco de exploração principal deste trabalho, algoritmos rápidos para a decisão da profundidade máxima da árvore de codificação foram desenvolvidos. Inicialmente, um algoritmo para predição inter-vistas da profundidade da árvore de codificação utilizando o padrão HEVC em modo *Simulcast* foi desenvolvido, o qual foi publicado em [30]. A seguir, uma nova solução para otimizar a decisão da profundidade máxima da árvore de CUs utilizando a extensão 3D-HEVC para vistas de textura, foi também apresentada.

Tais algoritmos foram propostos com base em análises das decisões de profundidade das árvores de CUs pertencentes a duas vistas adjacentes, com o intuito de definir a profundidade máxima das árvores de CUs a serem codificadas em uma terceira vista vizinha, reduzindo assim a complexidade computacional da codificação intra multivista.

1.3.3. Redução da Complexidade de Codificação Intra das Vistas de Textura e Profundidade do 3D-HEVC

Os avanços em tecnologias de display e a crescente popularidade de sistemas de vídeo em 3D, tem atraído cada vez mais consumidores para experiências com visualização de

vídeos em 3D e, conseqüentemente, a demanda por armazenamento e transmissão de conteúdo 3D tem aumentado consideravelmente. Para lidar com esta demanda, uma extensão 3D do padrão HEVC está sendo desenvolvida e se encontra em fase final de padronização.

Espera-se que o padrão 3D-HEVC proporcione maior eficiência de codificação do que seus antecessores, suportando múltiplas vistas com alta resolução e mapas de profundidade associados, a custo de um aumento considerável na complexidade computacional.

Nesta tese são apresentados métodos de aceleração da codificação intra tanto das vistas de textura quanto dos mapas de profundidade, com o objetivo de reduzir o tempo de processamento do codificador 3D-HEVC e possibilitar o seu uso em dispositivos portáteis com poder computacional limitado, tais como câmaras de vídeo 3D e *smartphones* equipados com várias câmaras e capacidade de aquisição de profundidade.

Os métodos propostos analisam a textura das tramas do vídeo e mapas de profundidade, visando estimar a orientação das bordas presentes nas PUs e reduzir o número de modos candidatos a serem avaliados na decisão do modo intra. Além disso, a correlação entre a escolha do modo de Planar como candidato e os MPMs (*Most Probable Modes*) selecionados é também explorada para otimizar a codificação dos mapas de profundidade. Os resultados desta exploração foram publicados em [31].

1.4. Organização da Tese

A estrutura do texto desta tese está organizada como segue.

No capítulo dois são apresentados alguns conceitos básicos e técnicas de codificação de dados de vídeo, as medidas de avaliação de desempenho aplicadas aos resultados obtidos neste trabalho, e a estrutura de codificação e principais ferramentas adotadas pelo novo padrão HEVC de compressão de vídeo.

No capítulo três, são apresentados alguns conceitos e aplicações de vídeo 3D, e logo após, as principais técnicas de codificação adotadas pela extensão 3D-HEVC, atualmente na fase final de padronização.

No capítulo quatro, são apresentadas as novas soluções algorítmicas desenvolvidas visando reduzir a complexidade da predição intra do HEVC, bem como os resultados alcançados e comparações com trabalhos relacionados.

No capítulo cinco, inicialmente, são apresentados os algoritmos desenvolvidos para a predição intra com inferência de modo inter-vista e, otimização da decisão da profundidade máxima da árvore de CUs, acompanhados dos resultados obtidos. Logo após, as inovações algorítmicas desenvolvidas para a redução de complexidade da codificação intratramas do 3D-HEVC são apresentadas, seguidas pelos resultados experimentais e comparações com trabalhos relacionados.

Finalmente, no capítulo seis, são apresentadas as conclusões do trabalho de pesquisa desenvolvido durante o doutoramento e apresentado nesta tese, bem como uma explanação sobre as contribuições realizadas e sugestões para trabalhos futuros.

2. CODIFICAÇÃO DE VÍDEO

Neste capítulo será apresentada uma revisão da literatura com alguns conceitos básicos sobre o assunto que esta tese abrange - codificação imagens e vídeo, e que serão importantes para a compreensão das soluções algorítmicas apresentadas no decorrer da tese. Inicialmente, será realizada uma contextualização do tema, com conceitos e descrição de algumas técnicas de compressão de dados de vídeo e, a seguir, uma revisão mais específica, focada no padrão HEVC.

2.1. Conceitos Básicos

Um sinal de vídeo é uma sequência de tramas, que possui uma taxa (ou frequência) definida pelo número de tramas exibidas por segundo. De modo que, quanto mais tramas por segundo (*fps*), mais suave o movimento aparece no ecrã. Em geral, aplicações de consumo típicas utilizam cerca de 30 *fps*. No entanto, altas taxas como 60 *fps* podem ser usadas em aplicações de tecnologia avançada, e taxas baixas como 15 *fps* podem ser usadas em videoconferências, por exemplo, quando a largura de banda de transmissão é baixa [32].

Cada trama é composta por uma matriz bidimensional de pixels, cujas dimensões indicam a resolução. Vídeos de alta definição (*FullHD*), por exemplo, possuem uma resolução de 1920x1088 pixels por trama, embora hajam diversas resoluções diferentes desenvolvidas de acordo com as necessidades do mercado.

Atualmente, a grande maioria das aplicações de vídeo digital usam vídeo a cores e por isso necessitam de um método de representação da informação de cor. Os valores do brilho (ou luminância) e da cor definem um vetor que representa o pixel em um espaço de cor [33].

No espaço de cor RGB, utilizado em ecrãs de computador, cada pixel é representado por três componentes que indicam as proporções relativas das cores: vermelho (*Red*), verde (*Green*) e azul (*Blue*). No espaço de cor RGB, geralmente essas três cores são armazenadas de forma igual, usando a mesma resolução para cada cor [33]. No entanto, tendo em vista que o sistema visual humano é mais sensível ao brilho

(luminância) do que à cor (crominância), é possível representar uma imagem de forma mais eficiente, subamostrando a informação de crominância e representando a luminância com uma resolução maior, sem sacrificar a qualidade de vídeo [32].

Explorando esta característica do sistema visual humano, o espaço de cor YCbCr tornou-se popular por permitir uma representação eficiente de imagens coloridas, na qual a informação de luminância (Y) e as informações de crominância vermelha (Cr) e azul (Cb) são transmitidas, e as duas últimas utilizando uma resolução menor do que a de luminância [33].

A representação YCbCr usualmente utiliza subamostragem das componentes de cor (Cb e Cr). Por exemplo no formato 4:2:0, para cada quatro pixels de luminância (Y) existe um pixel de Cb e um de Cr [34], conforme ilustrado na Figura 2.1(a). No entanto, outros formatos como 4:2:2 e 4:4:4 também podem ser utilizados.

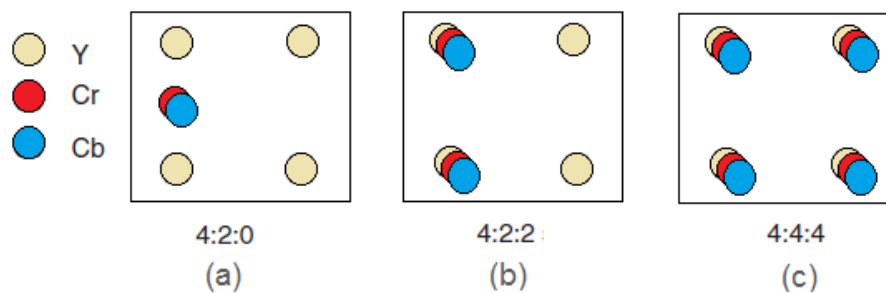


Figura 2.1: Exemplo de formatos de subamostragem de cores [33].

No formato 4:2:2 para cada quatro informações de Y, há duas informações de Cb e duas de Cr, conforme apresentado na Figura 2.1 (b). Já no formato 4:4:4, os componentes de crominância são totalmente preservados, como mostra a Figura 2.1 (c). O padrão HEVC suporta todos os formatos de crominância citados acima [4].

2.2. Técnicas de Compressão de Dados de Vídeo

A captura de um vídeo de alta definição em tempo real gera uma quantidade muito elevada de dados e seria inviável transmitir e/ou armazenar este vídeo se técnicas de compressão de dados específicas para vídeo não fossem empregadas.

O objetivo da compressão de vídeo é permitir o armazenamento e transmissão de dados visuais de forma eficiente. Neste caso, a quantidade de dados visuais deve ser

reduzida de tal forma que o tamanho do ficheiro final seja o menor possível sem comprometer a qualidade [35].

A compressão de dados de vídeo é alcançada através da remoção de redundância e da eliminação de informação visualmente irrelevante, ou seja, de informação que não é perceptível para a generalidade dos observadores. Muitos tipos de dados contêm redundância estatística e podem ser efetivamente comprimidos usando compressão sem perdas, de modo que os dados reconstruídos na saída do decodificador são uma cópia perfeita dos dados originais. No entanto, a compressão de vídeo sem perdas permite apenas uma quantidade moderada de compressão. De modo que, a compressão com perdas é necessária para que taxas de compressão mais altas sejam alcançadas [33].

Existem basicamente três tipos de redundâncias exploradas durante a compressão de vídeo: redundância temporal, redundância espacial e redundância estatística. A remoção desses quatro tipos de redundância e a redução de irrelevância visual durante a compressão de dados de vídeo são descritas a seguir.

- *Redundância temporal*: Esta redundância existe no domínio temporal, onde geralmente há uma alta correlação ou semelhança entre tramas temporalmente adjacentes, especialmente se a taxa de amostragem temporal ou taxa de tramas é alta [33]. A redundância temporal é explorada no processo de compressão recorrendo à predição temporal eventualmente melhorada com compensação de movimento (MC) inter-tramas usando informação de movimento obtida por um processo de estimação de movimento (ME) [35].

- *Redundância espacial*: Esta redundância entre pixels espacialmente próximos pode ser explorada usando técnicas de compressão de domínio espacial, referidas como codificação intra-tramas [35]. No domínio espacial, geralmente há uma alta correlação entre pixels (amostras) que são próximos uns dos outros, ou seja, os valores de amostras vizinhas muitas vezes são muito semelhantes [33]. A correlação espacial existente no sinal residual gerado com recurso a predição temporal inter-tramas com compensação de movimento é, adicionalmente, removida através do uso de transformadas (como a Transformada Discreta do Cosseno - DCT) [36].

- *Redundância Estatística*: As técnicas de redução da redundância estatística mais comuns em métodos para compressão de vídeo são a codificação de comprimento variável (*Variable Length Coding* - VLC) e a codificação aritmética (*Arithmetic Coding* -

AC). Nos codificadores padronizados são usadas variantes da codificação de *Huffman* e da codificação aritmética, sendo exemplos a codificação de comprimento variável com adaptação ao contexto (CAVLC) e a codificação aritmética binária com adaptação ao contexto (CABAC) usadas no padrão H.264/AVC [35].

- *Irrelevância Visual*: Em um sistema de compressão com perdas, os dados descompactados não são idênticos aos dados de origem e taxas de compressão maiores podem ser alcançadas à custa de uma perda de qualidade visual [33]. Felizmente, tirando partido das características da irrelevância visual de parte do conteúdo do sinal de vídeo, podemos comprimir os dados por duas ordens de magnitude sem prejudicar a qualidade do vídeo descomprimido. Esta compressão adicional consegue-se em geral através do descarte e representação mais grosseira de componentes espectrais (espaciais) de frequência mais elevada que ocorrem durante a quantização. Uma vez que o sistema visual humano é menos sensível às altas frequências o passo de quantização para os coeficientes de frequência elevada pode ser maior, sem afetar a qualidade visual [32].

2.3. Critérios de Avaliação

Para avaliar de forma adequada os resultados obtidos após a utilização de um algoritmo de codificação de vídeo, é importante definir alguns critérios de comparação.

Existem dois tipos principais de medidas para avaliar qualidade de um vídeo: subjetivas e objetivas [37]. A qualidade subjetiva não será discutida neste projeto, pois não foi utilizada nas avaliações apresentadas.

A medição da qualidade objetiva é realizada trama a trama, comparando os pixels das tramas do vídeo original com os pixels das tramas correspondentes pertencentes ao vídeo reconstruído (após descodificação).

O critério de avaliação mais utilizado para comparar a qualidade objetiva de um vídeo é o PSNR (*Peak Signal-to-Noise Ratio*) [38], o qual representa o grau de diferença entre a trama original e a trama reconstruída, usualmente considerando apenas a informação de luminância.

O PSNR está definido em (2.1), onde MAX é o valor máximo que uma amostra de luminância do vídeo pode atingir e MSE (*Mean Squared Error*) é o erro médio quadrático correspondente à diferença entre a trama original e a trama reconstruído, o qual será definido no decorrer desta seção.

$$PSNR = 20 \log \left(\frac{MAX}{\sqrt{MSE}} \right) \quad (2.1)$$

Em geral valores altos de PSNR indicam uma alta qualidade do vídeo codificado (e reconstruído) e os valores baixos indicam uma baixa qualidade. No entanto, o PSNR possui algumas limitações, pois existem casos em que uma imagem que possui uma qualidade visual ou subjetiva superior pode ter um valor de PSNR menor do que outra que possui uma qualidade visual inferior [37]. Tal fato ocorre porque o PSNR considera apenas o valor de diferenças quadráticas entre valores de pixels em blocos independentes da trama e deste modo, não avalia necessariamente os critérios subjetivos de qualidade [37].

O grau de similaridade entre duas tramas pode ser medido com base em diversas funções. Neste projeto, as duas principais funções de similaridade usadas foram o MSE, que também é usado na definição do PNSR, e o SAD (*Sum of Absolute Differences*).

Em (2.2), é apresentada a equação que define o MSE, na qual m e n representam as dimensões da trama a ser comparada e O e R correspondem às amostras da trama original e da trama reconstruída, respectivamente.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R_{i,j} - O_{i,j})^2 \quad (2.2)$$

O cálculo do SAD, definido em (2.3), avalia a distorção entre as regiões comparadas, a partir do somatório das diferenças absolutas, para cada ponto da trama original e da trama reconstruída [39].

Tal como em (2.2), na equação (2.3), m e n representam as dimensões da trama a ser comparada, O corresponde às amostras da trama original e, R corresponde às amostras da trama reconstruída.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |R_{i,j} - O_{i,j}| \quad (2.3)$$

Recentemente foi introduzida uma nova medida de avaliação de desempenho de codificação, o Bjøntegaard-Delta (*BD-PSNR/Rate*) [40][41], adequada à comparação de desempenho de dois métodos ou configurações de codificação por comparação de dois conjuntos de pontos de taxa-distorção (*Rate-Distortion - RD*). Tal método calcula o

desempenho que é representado por um de dois indicadores relacionados entre si. Um dos indicadores representa o valor médio de economia da taxa de bits, ou seja, o delta médio da taxa de bits para um conjunto de QPs (*Quantization Parameters*), sendo designado por *BD-Rate*. Um *BD-Rate* negativo indica um algoritmo de melhor desempenho, uma vez que corresponde a uma taxa de bits inferior a uma mesma qualidade. O outro indicador designado por *BD-PSNR* mede o acréscimo médio de PSNR para um conjunto de QPs, e se for positivo indica um ganho de desempenho.

Para o cálculo destes indicadores são considerados dois conjuntos de dados (original e proposto), cada um consistindo de 4 pares de valores (*PSNR*, *Bitrate*), um para cada QP. O PSNR já está em forma logarítmica. Usa-se também o logaritmo da taxa de bits, resultando em 4 pares de (*PSNR*, $\log(\text{Bitrate})$). Na Figura 2.2 é apresentado um exemplo de curvas e áreas usadas para definir os indicadores de desempenho de Bjøntegaard-Delta onde as áreas sombreadas representam na Figura 2.2 (a) o integral da diferença de PSNRs e na Figura 2.2 (b) o integral da diferença de taxas de bits. As áreas são limitadas para evitar a utilização de extrapolações das curvas para fora dos pontos finais. Os Bjøntegaard-Deltas (*BD-PSNR* e *BD-Rate*) são calculados dividindo essas áreas pela extensão do intervalo da base, representando por isso incrementos médios de PSNR e de taxa de bits de um método ou configuração de codificação relativamente ao outro.

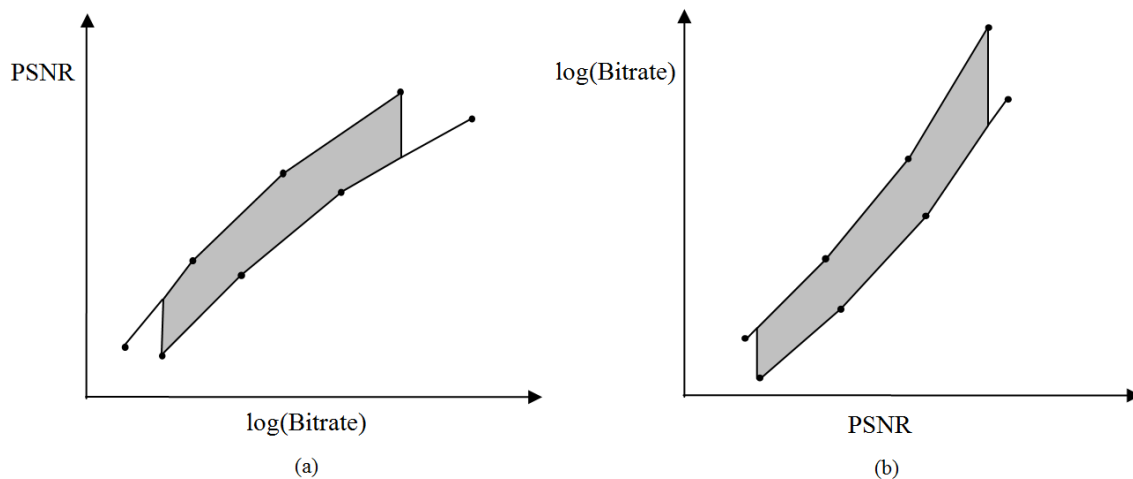


Figura 2.2: Exemplo de curvas de Bjøntegaard-Delta [41].

De forma simplificada o *BD-PSNR* avalia a média das diferenças de PSNR ao longo de todas as taxas de bits entre as curvas RD de dois algoritmos (original e proposto), de modo que um único valor é gerado a partir dos valores de teste utilizados. Da mesma

forma o *BD-Rate* avalia a média das diferenças na taxa de bits ao longo dos valores de PSNR apresentados entre as curvas RD dos algoritmos que estão sendo avaliados.

2.4. O Padrão HEVC

Com os avanços na tecnologia de codificação de vídeo, surgiu na indústria de comunicações a necessidade de um padrão para a representação de vídeo comprimido que fornecesse substancial aumento na eficiência de codificação de vídeo de alta e ultra alta resolução (resoluções HD e UHD), maior robustez a erros de transmissão e funcionalidades que facilitassem a paralelização [4].

Em resposta a essa crescente necessidade de maior compressão de vídeo para diversas aplicações, tais como videoconferência, mídia de armazenamento digital, televisão e internet streaming, em Abril de 2010 as atividades de desenvolvimento do padrão HEVC (*High Efficiency Video Coding*) foram iniciadas pelo grupo JCT-VC (*Joint Collaborative Team on Video Coding*), formado por especialistas do VCEG (*Video Coding Experts Group*) da ITU-T e do MPEG (*Moving Picture Experts Group*) do ISO/IEC. O esforço de padronização do HEVC teve como objetivo criar um algoritmo de codificação de vídeo capaz de aumentar a taxa de compressão em relação aos padrões existentes, sem afetar a qualidade do vídeo [5]. Três anos depois, em Abril de 2013, o padrão HEVC foi finalizado e padronizado com êxito [4].

Durante o período de desenvolvimento do HEVC e, mesmo após sua padronização, várias atividades de investigação foram e continuam a ser realizadas na busca por um maior equilíbrio entre a eficiência de compressão alcançada e a complexidade computacional exigida pelo padrão. A base de testes que permite essas atividades de pesquisa, é uma implementação referência, em software, do codificador e decodificador, disponibilizada e gerida pelo JCT-VC, designada *HEVC Test Model* ou HM [42]. No HM estão incluídas as novas ferramentas de codificação do padrão, podendo novos métodos serem incluídos, testados e avaliados utilizando sobre o HM. As ferramentas e a estrutura de codificação do novo padrão HEVC serão descritos em detalhe nas próximas seções.

2.4.1. Estrutura de Codificação

O HEVC utiliza a mesma estrutura de codificação híbrida baseada em um particionamento da trama a codificar em blocos, adotada em padrões anteriores como o H.264/AVC [18][44], combinando técnicas de predição temporal e transformadas matemáticas para reduzir a redundância no sinal de vídeo (ex.: predição com compensação de movimento para redução da redundância temporal e transformadas DCT para redução da redundância espacial) [35]. No entanto, diferente dos padrões de vídeo existentes, o HEVC utiliza uma estrutura de codificação baseada em árvores quaternárias com particionamento de blocos flexível realizado pela introdução de quatro diferentes conceitos de bloco, nomeadamente CTU, CU, PU, e TU [18], os quais são descritos em detalhe a seguir.

No HEVC, cada trama do vídeo de entrada é dividido em blocos quadrados de igual tamanho, chamados CTUs (*Coding Tree Units*). Cada CTU consiste de um bloco $N \times N$ de amostras de luminância, acompanhado de dois blocos de amostras de crominância correspondentes, de modo que as dimensões dos blocos de amostras de crominância vão depender da sub-amostragem de cores utilizada. O conceito de CTU é largamente análogo ao conceito de macrobloco em padrões anteriores, tais como H.264/AVC. As dimensões de uma CTU podem variar de 16×16 até 64×64 amostras de luminância [44]. Cada CTU pode ser composta de apenas uma unidade de codificação (CU - *Coding Unit*) ou pode ser dividida em múltiplas CUs, usando uma estrutura de particionamento em árvore quaternária (*quadtree*) [45], na qual as CTUs são as raízes de cada árvore de codificação.

2.4.1.1. Unidade de Codificação (CU)

A unidade de codificação (CU) é um bloco quadrado e seu tamanho pode variar de 8×8 pixels até o tamanho da CTU [44]. A CU é a unidade onde o tipo de predição a ser usado inter/intra é especificado [18].

Cada CU pode ser dividida recursivamente em quatro blocos de igual tamanho a partir do tamanho da CTU até o tamanho mínimo da CU ser atingido (8×8 pixels).

Na Figura 2.3, um exemplo de estrutura de árvore de CUs usando uma trama da sequência Flamenco2 é apresentado, no qual cada CTU 64×64 corresponde a uma CU 64×64 que pode ser particionada em CUs menores. Neste exemplo a árvore de codificação apresenta quatro níveis de profundidade, onde os nós-folha representam as CUs. A partir desta figura é possível notar que o número de árvores possíveis a serem testadas no

processo de otimização da taxa-distorção (*RDO - Rate-Distortion Optimization*) aumenta exponencialmente de acordo com a profundidade máxima da árvore.

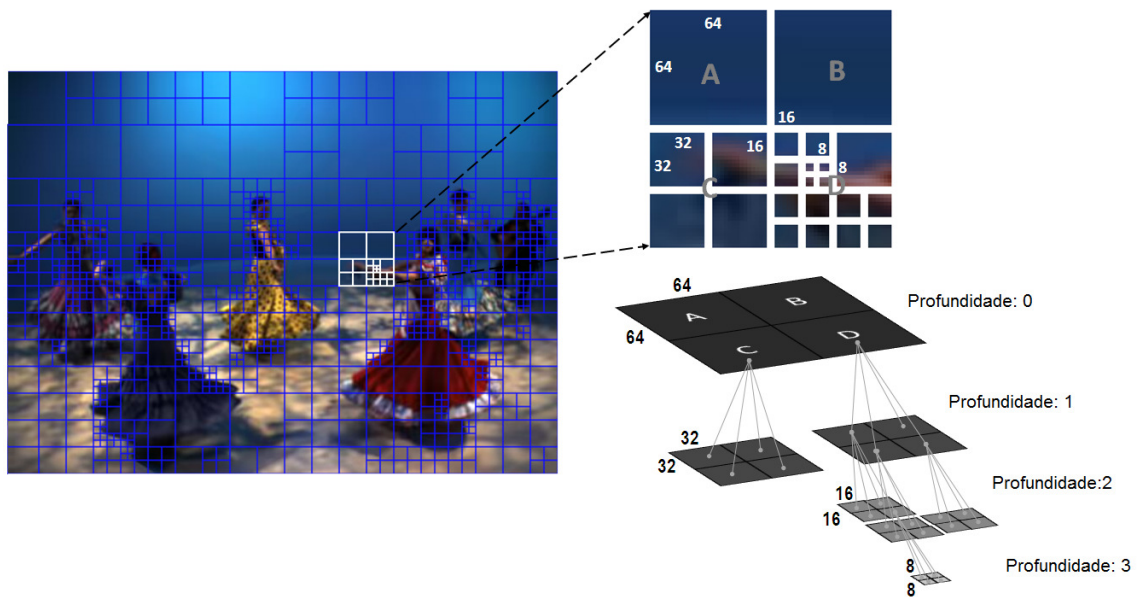


Figura 2.3: Exemplo de estrutura de codificação em árvore quadrática.

2.4.1.2. Unidade de Predição (PU)

A unidade de predição, chamada de *Prediction Unit* (PU) é a unidade espacial básica relacionada com os processos de predição. Em geral, não é restrita a ser de formato quadrado, podendo ter também formato retangular, a fim de melhorar a correspondência com os limites dos objetos reais da imagem. Cada CU pode conter uma, duas ou quatro PUs, sendo suportadas partições simétricas $2N \times 2N$, $N \times 2N$, $2N \times N$ e $N \times N$, e assimétricas $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$ [44], apresentadas na Figura 2.4. Nesta figura, as dimensões nU e nD representam a altura da partição superior (*Upper*) e inferior (*Down*) da PU, respectivamente, e as dimensões nL e nR representam a largura das partições à esquerda (*Left*) e à direita (*Right*).

Cada PU pode ser tão grande quanto uma CU ou tão pequeno quanto um bloco de tamanho 8×4 ou 4×8 pixels. A partição $N \times N$ só é permitida quando o tamanho da CU correspondente não é o menor permitido (8×8 pixels).

2.4.1.3. Unidade de Transformação (TU)

A unidade espacial de transformada, chamada de *Transform Unit* (TU) é a unidade básica utilizada para os processos de transformação matemática e quantização. A TU é uma região quadrada definida de acordo com o particionamento da CU.

O tamanho da TU pode variar de 4x4 à 32x32 amostras de luminância. Cada CU pode conter uma ou mais TUs, de modo que múltiplas TUs podem ser dispostas em uma estrutura de *quadtree*, conforme ilustrado na Figura 2.4. Uma vez que a transformada matemática é aplicada a um bloco de resíduos (após predição e subtração da referência) a estrutura formada a partir da partição do bloco em TUs é chamada de "residual *quadtree*" ou RQT [44]. Na Figura 2.4, um exemplo de particionamento de CU em múltiplas TUs é apresentado, no qual uma CU 32x32, destacada em cinza, é dividida em TUs de tamanho 16x16, 8x8 e 4x4.

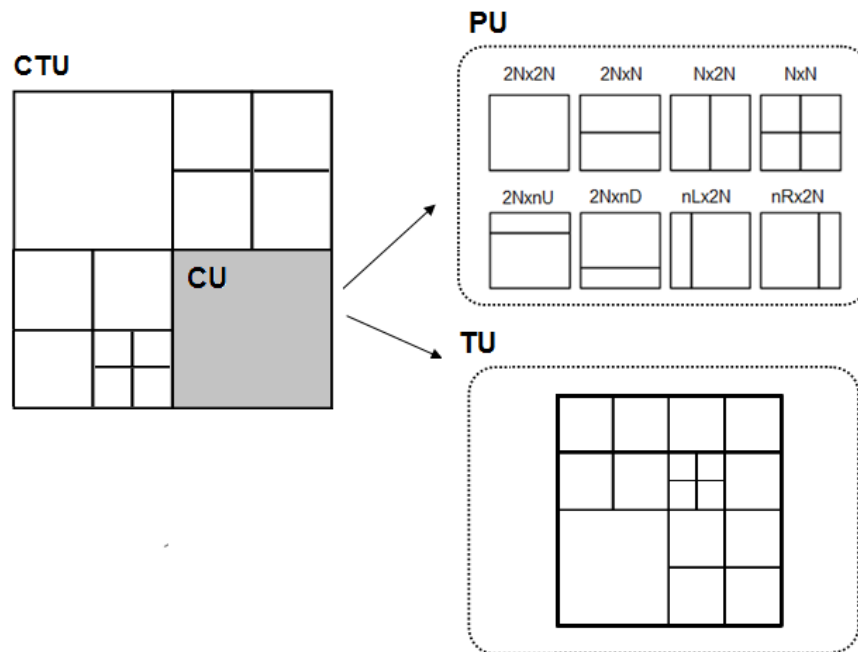


Figura 2.4: Estrutura de Codificação: Unidades de Codificação (CU), Predição (PU) e Transformação (TU).

2.4.1.4. Estrutura de Slices e Tiles

Uma *slice* é especificada como uma unidade de empacotamento de dados de vídeo codificados para efeitos de transmissão. Uma trama pode conter uma ou mais *slices*, de modo que cada *slice* é composta por um cabeçalho de informações seguido por uma sequência de CTUs processadas em *raster scan order* [5].

No HEVC uma *slice* pode ser composta por um ou mais segmentos de *slices*, os quais podem ser independentes ou dependentes. Inicialmente, um segmento de *slice* independente é processado, seguido por zero ou mais segmentos de *slice* dependentes.

Os segmentos independentes são semelhantes às *slices* usadas nos padrões anteriores como o H.264/AVC, são segmentos auto-contidos e podem ser decodificados independentemente, de modo que nenhuma previsão é realizada através das fronteiras do segmento de *slice* independente. Já os segmentos de *slice* dependentes permitem troca de informações entre segmentos tanto para o processo de decodificação de entropia quanto para o processo de reconstrução [46].

Os segmentos dependentes foram criados com o intuito de auxiliar na granularidade de empacotamento e transmissão de dados [5]. De modo que, após a transmissão de um segmento de *slice* independente, um segmento dependente codificado já pode ser transmitido, enquanto o próximo segmento dependente está a ser codificado [46]. A Figura 2.5 apresenta um exemplo da estrutura de codificação dos segmentos de *slice* independentes e dependentes.

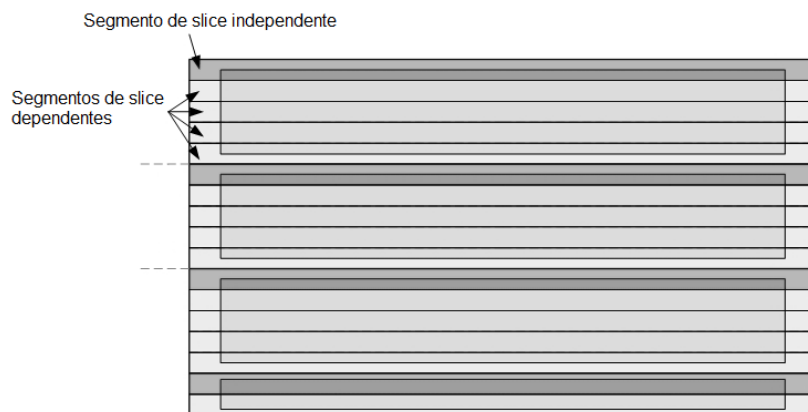


Figura 2.5: Exemplo de estrutura de codificação de segmentos de *slice* independentes e dependentes [46].

Além do conceito de *slices*, o padrão HEVC também define as *tiles*, as quais são regiões retangulares compostas por um conjunto de CTUs. De modo que as *tiles* são sempre alinhadas com os limites das CTUs. Uma *tile* pode ser composta por CTUs pertencentes a mais de uma *slice*, ou ainda, uma *slice* pode conter CTUs contidas em mais de uma *tile* [33], conforme ilustrado na Figura 2.6.

As *tiles* foram criadas com o objetivo de auxiliar no uso de arquiteturas de processamento paralelo para o codificador e decodificador [5][47]. As *tiles* são decodificadas de forma independente, de modo que o decodificador pode processar *tiles*

em paralelo com outras tiles. As CTUs dentro de uma *tile* são processadas em *raster order scan*, e as tiles dentro de uma trama também são processadas em *raster order scan*. Para facilitar essa decodificação em paralelo, a localização das *tiles* da trama deve ser indicada no *bitstream*. Em relação as *slices*, as tiles tem a vantagem de não conter cabeçalhos para a melhoria da eficiência de codificação [47].

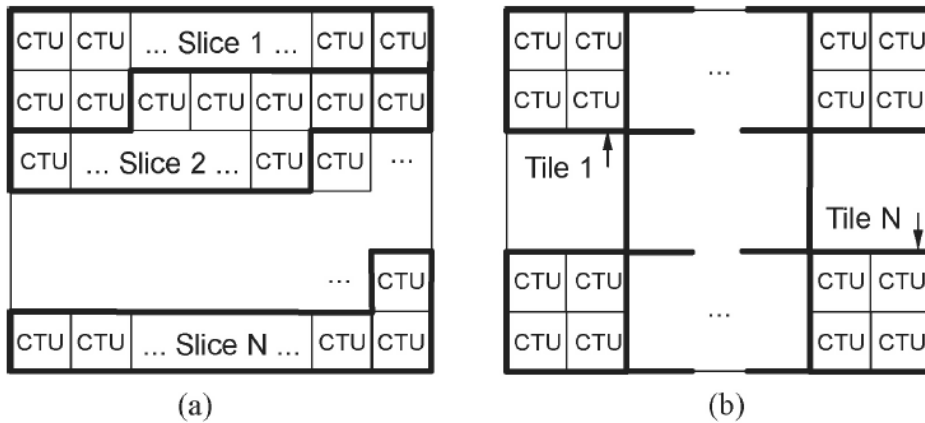


Figura 2.6: Exemplo de particionamento de uma trama em Slices (a) e em Tiles (b) [5].

2.4.2. Ferramentas de Codificação

A seguir são apresentadas as ferramentas de codificação que compõem o novo padrão HEVC. A Figura 2.7 apresenta um diagrama em blocos do codificador para que seja possível visualizar a localização de cada uma das ferramentas no processo de codificação.

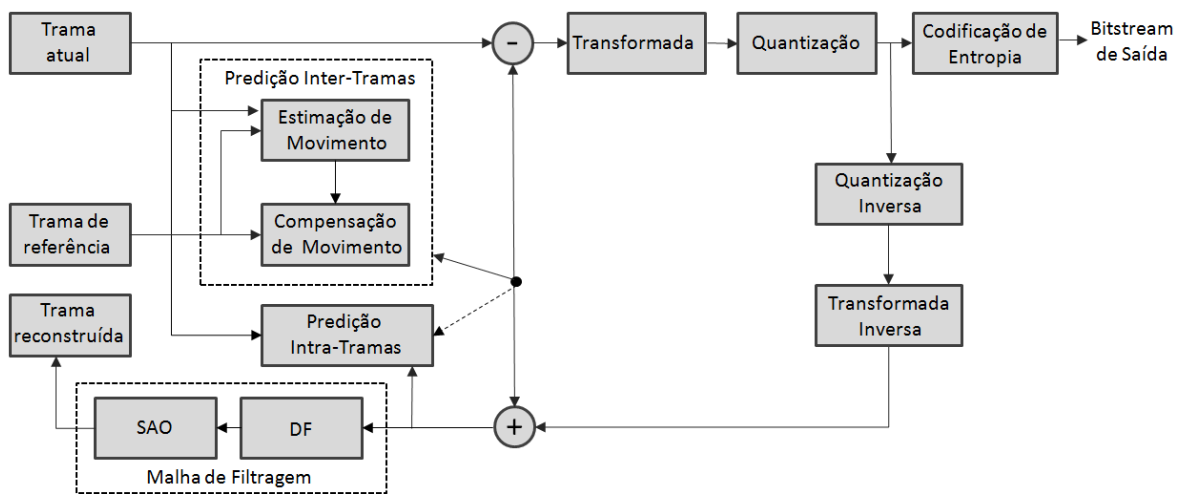


Figura 2.7: Diagrama em blocos de um codificador HEVC.

2.4.2.1. Predição Intra-Trama

A ferramenta de predição intra do HEVC utiliza um conjunto de 33 modos de predição angulares (direcionais), conforme apresentado na Figura 2.8, além dos modos DC e Planar [5]. O modo com melhor desempenho entre esses 35 é identificado através de um procedimento de busca exaustiva que avalia todos os modos e seleciona aquele que apresentar maior eficiência de codificação, considerando-se os valores da taxa de distorção (RD). Tal processo será apresentado em mais detalhe no Capítulo 4.

A Figura 2.8, apresenta os 33 modos intra angulares usados no HEVC, de VER-8 a VER+8, e de HOR-7 a HOR+8. Nesta figura, os pixels de referência cujos valores serão usados para formar a predição do pixel a codificar são destacados em amarelo e consistem das amostras das PUs vizinhas já codificadas acima e à esquerda da PU a ser processada.

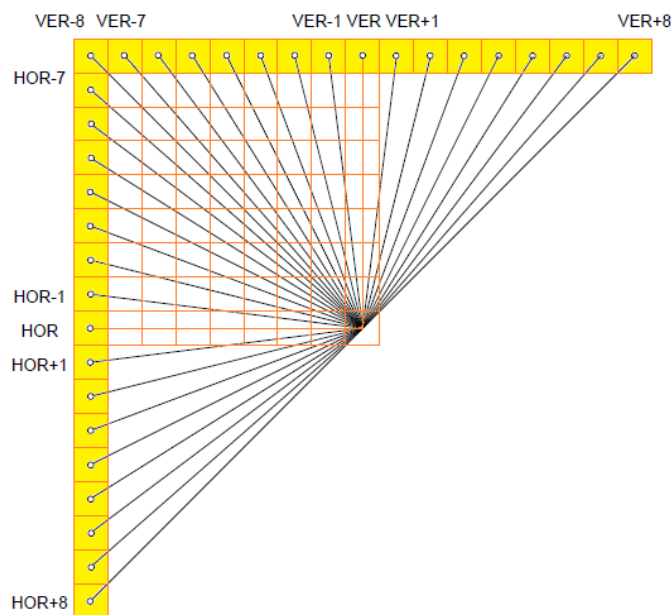


Figura 2.8: Direções de predição intra no HEVC [49].

Para as amostras preditas usando o modo DC, a predição é o valor da média aritmética das amostras que compõem as bordas esquerda e superior da PU a ser predita. A predição usando o modo Planar é concebida para ser capaz de reconstruir segmentos mais homogêneos da imagem, permitindo a continuidade máxima do plano da imagem, de modo a evitar descontinuidades nas fronteiras da PU [5]. Quando uma PU é predita usando o modo Planar sua amostra inferior direita é sinalizada no fluxo de bits e as amostras na coluna mais à direita e na coluna inferior da PU são interpoladas linearmente, conforme

apresentado na Figura 2.9 (a). As amostras centrais do sinal de predição são interpoladas bi-linearmente a partir das amostras de fronteira do sinal reconstruído [5][49], conforme apresentado na Figura 2.9 (b). Quando o modo Planar é sinalizado, o mesmo algoritmo é aplicado para ambos os componentes de luminância e crominância. O modo Planar é suportado por todos os tamanhos de bloco no HEVC, enquanto no H.264/AVC tal modo é suportado somente por blocos de 16x16 amostras de luminância [5].

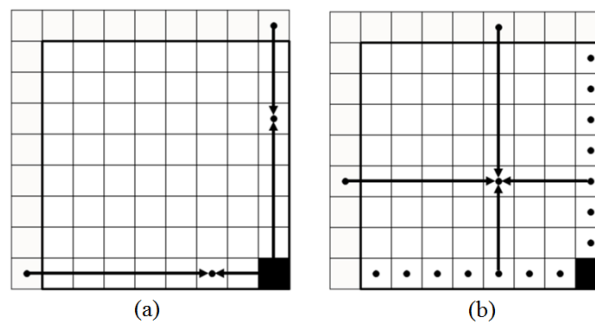


Figura 2.9: Predição planar de uma PU 8x8, amostras interpoladas linearmente (a) e bi-linearmente (b) [49].

O mapeamento entre as direções de predição intra e os números dos modos intra é apresentado na Figura 2.10. Os números das setas correspondem ao número pelo qual o modo de predição angular é referido.

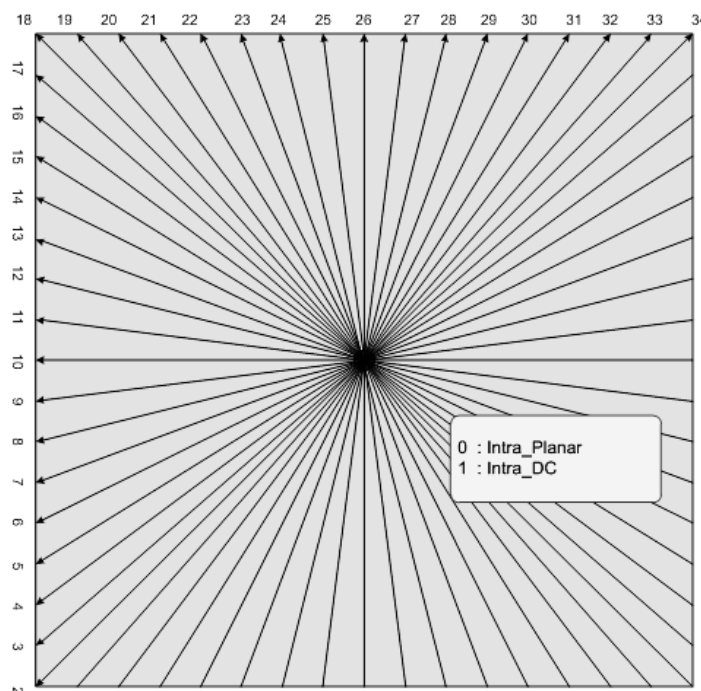


Figura 2.10: Mapeamento entre a direção e o modo de predição intra [44].

Um descrição mais detalhada do processo de Predição Intra-trama do HEVC é apresentada no Capítulo 4 deste tese.

2.4.2.2. Predição Inter-Trama

As PUs em que a predição tem por base tramas diferentes após compensação de movimento são acompanhadas por um conjunto de parâmetros de movimento que consistem em vetores de movimento, índices das tramas de referência, e índices de uso das listas de tramas de referência a ser utilizadas para geração das amostras preditas. Estes parâmetros são sinalizados de forma explícita ou implícita [44].

Quando uma CU é codificada com o modo SKIP, ela é composta por apenas uma PU, a qual não possui coeficientes de transformada, vetores de movimento ou índice da trama de referência. No modo *merge*, que pode aparecer associado ao modo SKIP ou a qualquer outro modo de predição inter-trama, os parâmetros de movimento da PU atual são obtidos a partir das PUs vizinhas [44].

O codificador tem a liberdade para usar o modo *merge* ou transmitir explicitamente os parâmetros de movimento, de modo que os vetores de movimento, índices da trama (ou tramas) de referência e índices de direção de predição sejam sinalizados explicitamente para cada PU. Assim, o modo *merge* tem a função de sinalizar que a informação de movimento de uma PU é obtida a partir da informação de movimento de PUs espacial ou temporalmente vizinhas.

O codificador pode selecionar os melhores parâmetros de movimento inferidos a partir de uma lista de candidatos formada pelas PUs espacialmente vizinhas (A_i , B_j apresentadas na Figura 2.11) e pelas PUs temporalmente vizinhas ou co-localizadas, transmitindo o índice correspondente que indicará o candidato escolhido.

Na Figura 2.11 são apresentadas as posições das PUs candidatas espacialmente vizinhas, a partir das quais a seguinte ordem de derivação é utilizada: $A_1 \rightarrow B_1 \rightarrow B_0 \rightarrow A_0 \rightarrow (B_2)$. De modo que a PU da posição B_2 é considerada somente se alguma das outras quatro PUs candidatas não estiver disponível ou for intra codificada.

Para as PUs candidatas temporalmente vizinhas, a posição inferior direita exterior à PU co-localizada na trama de referência é usada se estiver disponível. Caso contrário, a PU da posição central da PU co-localizada é usada em seu lugar. A lista de tramas de referência a ser utilizada para a derivação da PU co-localizada é explicitamente sinalizada no cabeçalho da *slice* [5][44].

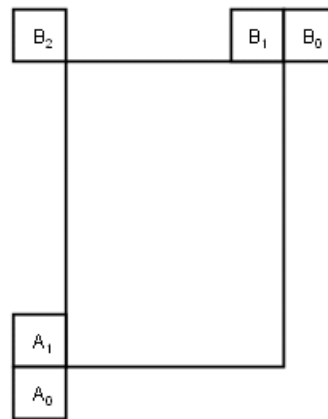


Figura 2.11: Posições das PUs espacialmente candidatas para o Modo Merge [44].

A técnica de Predição do Vetor de Movimento (*Motion Vector Prediction*), a qual explora a correlação espaço-temporal do vetor de movimento com as PUs vizinhas, é usada quando os parâmetros de movimento são transmitidos explicitamente. Tal técnica constrói uma lista de vetores de movimento candidatos através da verificação da disponibilidade das PUs espacialmente vizinhas à esquerda e acima, bem como das PUs temporalmente vizinhas, removendo os candidatos redundantes (repetidos) da lista. O codificador seleciona então como vetor de movimento para a PU, o melhor vetor de entre os da lista de candidatos gerada e transmite o índice correspondente que indica o candidato escolhido.

Tendo em vista que a diferença entre duas tramas de um vídeo nem sempre corresponde a posições inteiras de pixel, o HEVC utiliza, como o H.264/AVC, compensação de movimento com precisão de fração de posição de pixel. Pixels não posicionados em posições inteiras são calculados usando filtros de interpolação.

Para as amostras de luminância, é aplicado um filtro separável de 7 *taps* baseado na DCT para as posições de 1/4 de pixel, e para as posições de 2/4 de pixel é usado um filtro separável de 8 *taps* baseado na DCT, cujos coeficientes se apresentam na Tabela 2.1. No caso das amostras de crominância para posições de 1/8 de pixel, é usado um filtro separável de 4 *taps* baseado na DCT, cujos coeficientes se apresentam na Tabela 2.2.

Tabela 2.1: Coeficientes do filtro de 7-taps e 8-taps para interpolação de posições de 1/4 e 2/4 de pixel para a componente de luminância [44].

Posição	Coeficientes do Filtro
1/4	{-1, 4, -10, 58, 17, -5, 1}
2/4	{-1, 4, -11, 40, 40, -11, 4, -1}
3/4	{1, -5, 17, 58, -10, 4, -1}

Tabela 2.2: Coeficientes do filtro de 4-taps para interpolação de posições de 1/8 de pixel para as componentes de crominância [44].

Posição	Coeficientes do Filtro
1/8	{-3, 60, 8, -1}
2/8	{-4, 54, 16, -2}
3/8	{-5, 46, 27, -4}
4/8	{-4, 36, 36, -4}
5/8	{-4, 27, 46, -5}
6/8	{-2, 16, 54, -4}
7/8	{-1, 8, 60, -3}

2.4.2.1. Transformada e Quantização

O padrão HEVC suporta transformadas de tamanhos 4x4, 8x8, 16x16 e 32x32, sendo o tamanho a usar dependente do tamanho da TU. As transformadas cujas matrizes são definidas no padrão [50] são então aplicadas aos blocos de resíduos de predição resultando em blocos de coeficientes das transformadas. Estes coeficientes são então quantizados visando reduzir a irrelevância visual contida nos resíduos [32] introduzindo-se no processo um forte factor de compressão dos dados.

A quantização reduz a precisão da representação dos coeficientes gerados pelas transformadas e, por esse motivo introduz no processo de codificação perdas de informação irreversíveis. A magnitude destas perdas é controlada pelo parâmetro de quantização (QP) definido na configuração do codificador (caso de QP constante e taxa de bits variável) ou por um algoritmo de controlo de taxa de codificação (QP variável para taxa de bits constante).

2.4.2.2. Codificação de Entropia

Após a quantização, a codificação de entropia é aplicada a todos os elementos sintáticos e coeficientes transformados quantizados.

No padrão H.264/AVC, precedente ao HEVC, dois métodos de codificação de entropia são definidos: o CAVLC (Codificação de Comprimento Variável Adaptativa ao Contexto) que é a base do codificador de entropia e o CABAC (Codificação Aritmética Binária Adaptativa ao Contexto) que é opcional e selecionável nos perfis *Main* e *High* [1]. O CABAC provê uma melhor eficiência de codificação que o CAVLC, pois possui codificação aritmética e uma modelagem de contexto mais sofisticada. No entanto, a utilização do CABAC aumenta de forma considerável a complexidade de codificação.

No HEVC, somente o método CABAC é usado para a codificação de entropia de todos os tipos de configurações do codificador. O núcleo do algoritmo CABAC usado no HEVC é essencialmente o mesmo usado no CABAC do H.264/AVC [51], mas com algumas modificações. O CABAC usado no HEVC utiliza um número consideravelmente menor de contextos em comparação com o H.264/AVC, mas consegue fornecer um desempenho de compressão ligeiramente melhor, e maior velocidade de processamento [5] [50].

No HEVC, além dos modelos de contexto obtidos a partir da informação das CUs espacialmente vizinhas, o particionamento para decisão da profundidade das árvores de CUs ou TUs é também explorado para gerar índices de modelos de contexto de vários elementos sintáticos. Além disso, no HEVC o modo de codificação *bypass* do CABAC é extensivamente usado, visando aumentar a taxa de processamento do codificador através da redução do volume de dados a serem codificados utilizando modelos de contexto [5].

2.4.2.3. Malha de Filtragem

A filtragem na malha de predição utilizada no HEVC é composta basicamente por dois estágios: Filtro de Redução de Efeito de Bloco (DF) e o Deslocamento Adaptativo à Amostra (SAO), os quais estão descritos em mais detalhe a seguir.

2.4.2.3.1. Filtro de Redução de Efeito de Bloco

O processo de filtragem do HEVC utilizando o Filtro de Redução de Efeito de Bloco (*Deblocking Filter* - DF), tem como base o filtro utilizado no H.264/AVC [2]. O

Deblocking Filter tem a função de suavizar os efeitos de bloco ocasionados pela quantização durante a codificação do vídeo.

No HEVC, a filtragem é realizada para cada CU com a mesma ordem utilizada no processo de decodificação, no qual as bordas verticais são filtradas primeiro (filtragem horizontal) e logo após as bordas horizontais são filtradas (filtragem vertical). A filtragem é aplicada às bordas dos blocos transformados de tamanho 8x8 ou maior, para ambos os componentes de luminância e crominância. As bordas de blocos 4x4 não são filtradas no padrão HEVC, a fim de reduzir a complexidade computacional.

A Figura 2.12, apresenta um exemplo de codificação sem aplicação do Filtro Redutor de Efeito de Bloco (Figura 2.12(a)) e, com a aplicação do filtro (Figura 2.12(b)).



Figura 2.12: Antes da aplicação do *Deblocking Filter*(a) e após a aplicação do *Deblocking Filter* (b).

2.4.2.3.2. Deslocamento Adaptativo à Amostra (SAO)

Após a aplicação do Filtro Redutor de Efeito de Bloco, um Deslocamento Adaptativo à Amostra (*Sample Adaptive Offset – SAO*) é aplicado para a reconstrução do sinal, usando os valores de deslocamento fornecidos pelo codificador para cada CTU. A localização do SAO no fluxo de codificação é apresentada na Figura 2.7.

O codificador decide a aplicação ou não do SAO na *slice* atual. De modo que, se o SAO for aplicado, os pixels reconstruídos que compõem cada CTU da *slice* são classificados em categorias conforme suas intensidades e, a distorção é reduzida adicionando-se um deslocamento, de banda (*Band Offset*) ou de borda (*Edge Offset*), aos pixels de cada categoria.

O deslocamento de banda classifica todos os pixels de uma região em diversas bandas, onde cada banda contém pixels num intervalo de mesma intensidade. E o deslocamento de borda classifica os pixels em categorias a partir da sua semelhança com os pixels vizinhos.

2.4.3. Estruturas de Predição Temporal

O codificador HEVC suporta três tipos de estruturas de predição temporal, que são classificadas como: Apenas Intra (*Intra-Only*), Baixo Atraso (*Low-Delay*) e Acesso Aleatório (*Random-Access*) [44]. O gerenciamento da lista de tramas de referência depende da configuração de estrutura temporal selecionada. A seguir estão descritas cada delas:

2.4.3.1. *Intra-only* ou *All-Intra* (AI)

Na configuração *Intra-only*, são admitidos somente tramas do tipo I, as quais não dependem de tramas anteriormente codificados, pois a predição intra acontece considerando apenas amostras da trama atual, deste modo não são utilizadas tramas de referência temporal e, portanto, não há predição temporal. Nesta configuração não é permitido alterar o valor do parâmetro de quantização (*Quantization Parameter - QP*) no decorrer de uma sequência de vídeo codificada.

A Figura 2.13 ilustra um exemplo de configuração *Intra-only*, apresentando uma sequência de tramas codificadas, onde o número associado a cada trama representa a ordem de codificação.

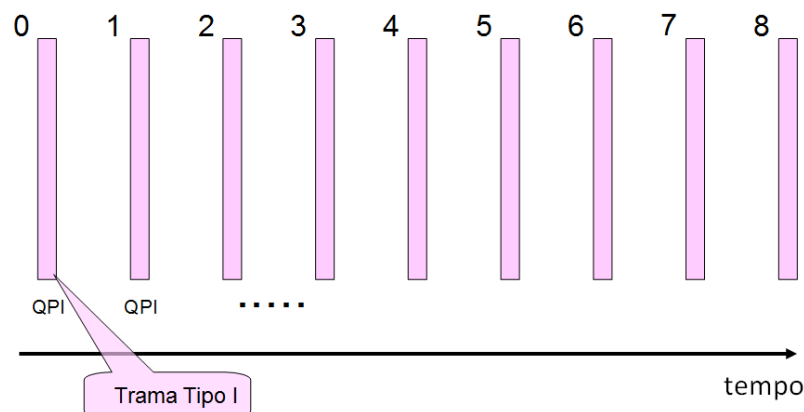


Figura 2.13: Sequência de vídeo codificada utilizando configuração *Intra-only* [44].

2.4.3.2. Low-delay (LD)

Na configuração *Low-Delay* apenas a primeira trama de uma sequência de vídeo deve ser codificada como uma trama tipo I. As tramas seguintes deverão ser codificadas como tramas do tipo GPB (*Generalized P and B Pictures*), as quais podem utilizar até duas tramas como referência na predição inter-tramas. No caso da configuração *Low-Delay*, o tipo GPB deve usar apenas tramas de referência que possuam POC (*Picture Order Count*) menor do que a trama atual, ou seja, todas as tramas de referência devem ser temporalmente anteriores na ordem de exibição em relação à trama atual. A Figura 2.14, apresenta um exemplo de sequência utilizando a configuração *Low-Delay*, onde o número associado a cada trama representa a ordem de codificação. É possível notar que a ordem de codificação é igual à ordem de exibição das tramas nesta configuração.

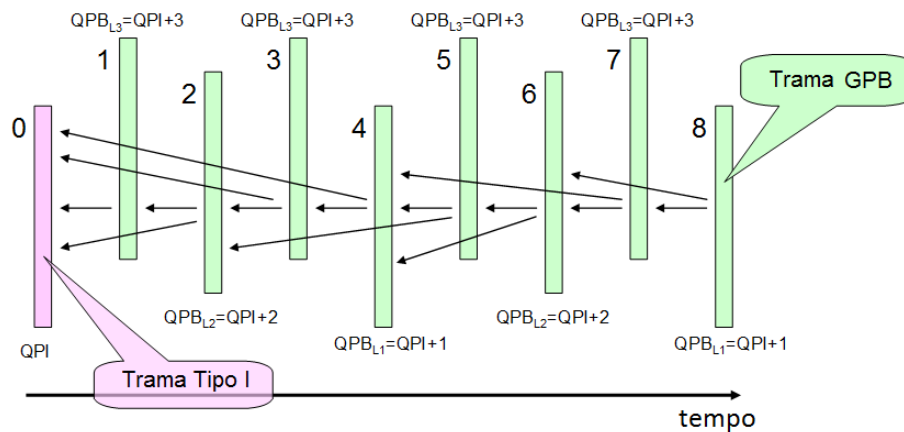


Figura 2.14: Sequência de vídeo codificada utilizando configuração *Low-Delay* [44].

2.4.3.3. Random-Access (RA)

Na configuração *Random-Access* uma estrutura de tramas B (*Bi-predictive*) hierárquica é utilizada na codificação. Nesta configuração, a primeira trama da sequência de vídeo é do tipo I e as tramas seguintes são do tipo B, nas quais a predição inter é gerada considerando, simultaneamente, amostras de duas tramas previamente codificadas, passadas ou futuras, na sequência de vídeo.

Uma nova trama tipo I é codificada ciclicamente em intervalos de cerca de um segundo, conforme o período intra definido na configuração do codificador, determinando o início de um novo GOP (*Group of Pictures*) [44]. Deste modo as tramas localizadas entre tramas intra sucessivas em ordem de exibição devem ser codificadas como tramas do tipo B. A Figura 2.15 mostra um exemplo de sequência de vídeo codificada utilizando

configuração *Random-Access*, onde a trama GPB usada na camada temporal mais baixa, não tem acesso às tramas das demais camadas temporais superiores e pode referenciar somente tramas I ou GPB já exibidos. Na segunda e terceira camadas temporais, as tramas do tipo B são codificadas com predição inter-tramas bidirecional e podem usar como referência quaisquer tramas do GOP das camadas temporais inferiores. Por fim, a camada temporal mais alta contém apenas tramas B não referenciadas por nenhuma trama do GOP, já que não há mais camadas a serem codificadas.

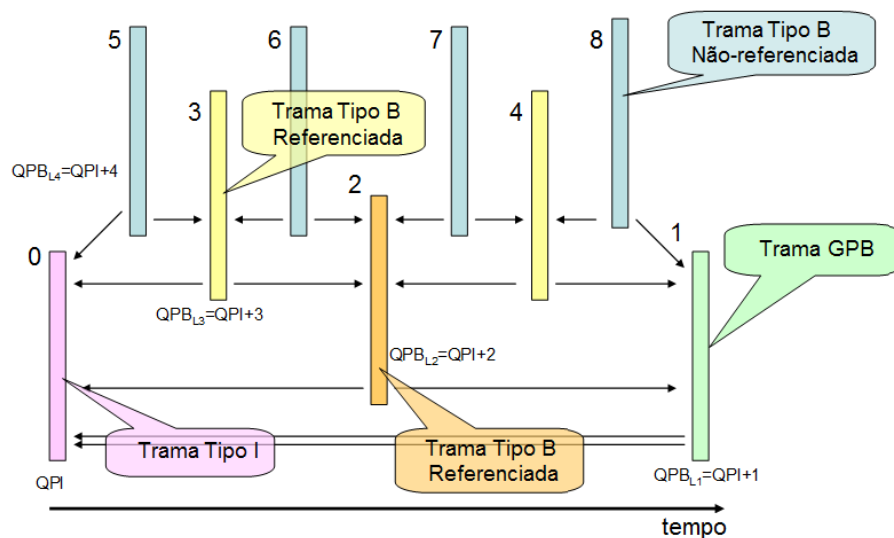


Figura 2.15: Sequência codificada utilizando configuração *Random-Access* [44].

Neste capítulo, além de alguns conceitos básicos sobre codificação de vídeo e técnicas de compressão de dados, foram apresentadas a estrutura de codificação do padrão HEVC e a descrição de suas principais ferramentas, visando facilitar a leitura e compreensão dos próximos capítulos desta tese. De entre essas ferramentas, a codificação intra-tramas é o foco principal desta tese e será apresentada em mais detalhe no Capítulo 4. Além disso, dado que na nova estrutura de particionamento adotada pelo HEVC as divisões possíveis de CUs em PUs de menor tamanho na árvore de codificação hierárquica influenciam a complexidade da codificação intra utilizada, as decisões de profundidade da árvore de codificação também são exploradas no decorrer deste trabalho.

Tendo em atenção que esta tese abrange contribuições para a redução de complexidade da codificação intra da extensão para vídeo 3D, 3D-HEVC, o capítulo a seguir apresenta algumas aplicações de codificação de vídeo 3D assim como a estrutura de codificação e novas técnicas adotadas nessa extensão.

3. CODIFICAÇÃO DE VÍDEO EM 3D

Este capítulo irá tratar da codificação de vídeo 3D, apresentando algumas das principais aplicações de sistemas de vídeo 3D, bem como uma descrição dos conceitos e novas ferramentas de codificação da extensão 3D-HEVC, a qual se encontra em fase final de padronização.

3.1. Sistemas de Vídeo 3D

Atualmente, a compressão e transmissão de vídeos estereoscópicos está a evoluir para sistemas mais avançados, chamados de sistemas de segunda geração, os quais fornecem ao espectador uma percepção mais realista do espaço 3D [52]. Tais sistemas necessitam de representações eficientes das cenas 3D, as quais são obtidas através da utilização do vídeo multivista juntamente com os mapas de profundidade correspondentes, estimados durante o processo de produção do conteúdo ou capturados juntamente com a textura.

Aplicações potenciais dos sistemas de vídeo 3D de segunda geração incluem principalmente ecrãs auto-estereoscópicos e televisões com ponto de vista livre (FTV - *Free viewpoint Television*):

- Ecrãs auto-estereoscópicos dispensam a necessidade de utilizar óculos especiais, pois fornecem uma percepção binocular de profundidade estereoscópica sem o uso de quaisquer aparelhos adicionais. Comumente, essa percepção ou sensação de profundidade é alcançada através de barreiras de paralaxe ou lentes, que permitem que cada olho do observador receba um conjunto de pixels diferente para cada perspectiva [54][55]. Os ecrãs ou televisores auto-estereoscópicos exibem simultaneamente múltiplas vistas da mesma cena 3D, as quais correspondem à cena vista de vários ângulos de visão ligeiramente diferentes. Ao exibir múltiplas vistas esses ecrãs criam um efeito de paralaxe dinâmica, ou seja, o ponto de vista do conteúdo exibido ao observador muda quando ele muda a sua posição.

- FTV é um sistema interativo que permite aos espectadores ou usuários experimentarem uma navegação com ponto de vista livre em uma cena 3D, podendo escolher o ponto de vista que desejarem dentro da faixa coberta pelas câmeras ou vistas. Tal tecnologia necessita de muitas vistas para proporcionar uma navegação (em termos de ponto de vista) fluída e livre de artefatos. Além disso, o receptor tem de sintetizar as vistas intermédias adicionais, de tal modo que o ponto de vista e a direção de observação possam ser alterados muito suavemente [56].

As aplicações mencionadas acima exigem uma transmissão eficiente do conteúdo de vídeo 3D que consiste de múltiplas vistas e seus mapas de profundidade. Como os mapas de profundidade não são diretamente observados pelo espectador, mas sim utilizados para sintetizar as vistas virtuais necessárias à exibição do vídeo em um ecrã auto-estereoscópico ou sistema FTV, a qualidade dos mapas de profundidade descodificados é expressa através da qualidade das vistas sintetizadas com o seu contributo.

3.2. A extensão 3D-HEVC

A crescente popularidade dos sistemas de vídeo 3D na indústria de cinema e aplicações de entretenimento, e a necessidade de fornecer múltiplas vistas para a operação de ecrãs auto-estereoscópicos e FTVs têm estimulado o desenvolvimento de novas tecnologias para codificação de vídeo multivista nos últimos anos.

Assim, para atender a esse crescimento na demanda por transmissão e/ou armazenamento de vídeo 3D, o grupo JCT-3V iniciou o desenvolvimento de uma extensão multivista para o padrão HEVC, denominada 3D-HEVC [11][12][13], a qual se encontra em fase final de padronização. As seções seguintes deste capítulo, apresentam as novas técnicas e métodos adotados pelo 3D-HEVC, bem como uma descrição detalhada da estrutura do sistema e processo de codificação.

3.2.1. Formato de Dados e Estrutura do Sistema

A extensão 3D-HEVC foi desenvolvida utilizando o formato de vídeo multivista mais profundidade (MVD) [12][57] com duas ou mais vistas, em que cada vista associa a cada pixel uma componente de textura e uma componente de profundidade. A

informação de profundidade por pixel é comumente codificada utilizando oito bits, onde o ponto mais próximo da câmera (z_{near}) é associado ao valor 255, e o ponto mais distante (z_{far}) é associado ao valor 0 [16], gerando uma trama de informação de profundidade na qual os tons de cinzento representam a distância da câmera até os objetos, conforme apresentado na Figura 3.1.



Figura 3.1: Representação do formato vídeo MVD com tramas de textura e mapas de profundidade associados [16].

De forma que, objetos mais próximos são representados por tons de cinzento mais claros e conforme os objetos se afastam da câmera, eles são representados por cores mais escuras. A equação (3.1) [58] representa a relação entre a profundidade z e o valor codificado no mapa de profundidade, d_r . Os valores z_{near} e z_{far} são as distâncias mínima e máxima dos objetos da cena [58].

$$\frac{1}{z} = \frac{d_r}{255} \cdot \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right) + \frac{1}{z_{far}} \quad (3.1)$$

Logo após a descodificação do vídeo e dados de profundidade, a renderização das vistas virtuais intermédias para a apresentação do conteúdo 3D em um ecrã auto-estereoscópico, ou de uma vista intermédia para apresentação num ecrã monoscópico, pode ser feita usando técnicas DIBR [59][15].

A estrutura do sistema de 3D-HEVC é descrita a seguir. As tramas de textura do vídeo e os mapas de profundidade codificados respeitantes a um instante temporal são agrupados em Unidades de Acesso (*Access Units*), como ilustrado Figura 3.2 [11]. A trama de textura e mapa de profundidade correspondente para uma posição de câmera particular são indicadas por um identificador da vista (*viewId* na Figura 3.2), o qual também é utilizado para especificar a ordem de codificação. A vista com o identificador igual a 0, também referida como vista base é codificada de forma independente das outras vistas usando um codificador de vídeo HEVC convencional. As demais vistas são chamadas de

vistas dependentes e são codificadas utilizando as ferramentas de codificação adicionais acrescentadas ao codificador base e que coletivamente constituem a extensão 3D-HEVC.

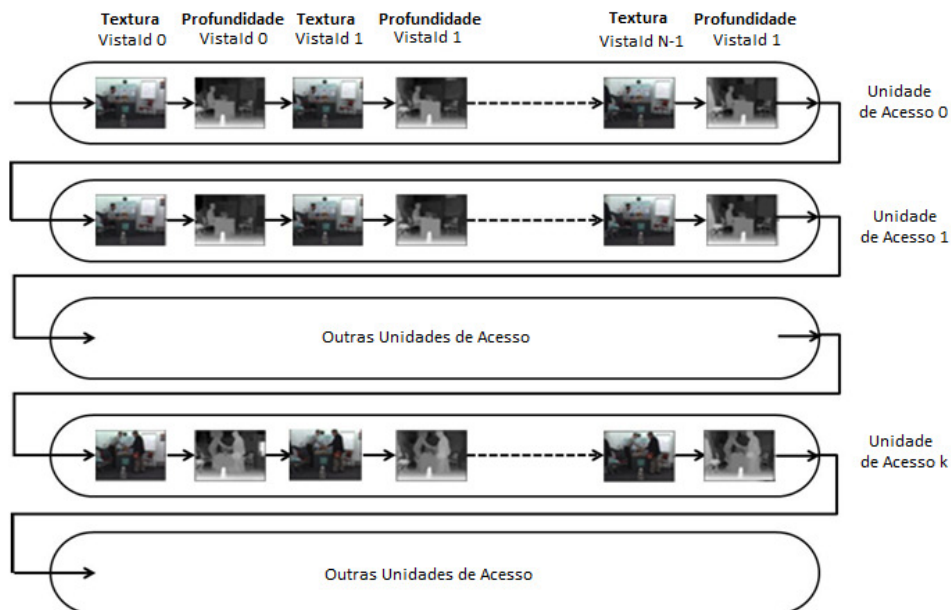


Figura 3.2: Estrutura das Unidades de Acesso e Ordem de Codificação dos Componentes de Textura e Profundidade [11][44].

3.2.2. Processo de Codificação

A estrutura do codificador de vídeo com extensão 3D-HEVC é apresentada na Figura 3.3 a um nível de abstração elevado. A fim de fornecer compatibilidade com sistemas de vídeo 2D, a vista base ou independente é codificada utilizando um codec totalmente compatível com o HEVC. Para codificar as vistas dependentes e mapas de profundidade, para além dos conceitos utilizados nos codecs HEVC, ferramentas de codificação adicionais e técnicas de predição inter-componentes são incluídas no 3D-HEVC, as quais utilizam dados já codificados em outras vistas da mesma unidade de acesso, conforme indicado pelas setas vermelhas na Figura 3.3.

O sub-bitstream contendo a vista independente pode ser decodificado por um decodificador de vídeo HEVC não modificado e exibido em um ecrã 2D convencional.

O codec 3D-HEVC também pode ser usado para a codificação de sinais de vídeo multivista sem dados de profundidade. Além disso, quando se utilizam dados de profundidade, a predição inter-componente pode ser configurada de forma que as tramas do vídeo possam ser decodificadas independentemente dos dados de profundidade.

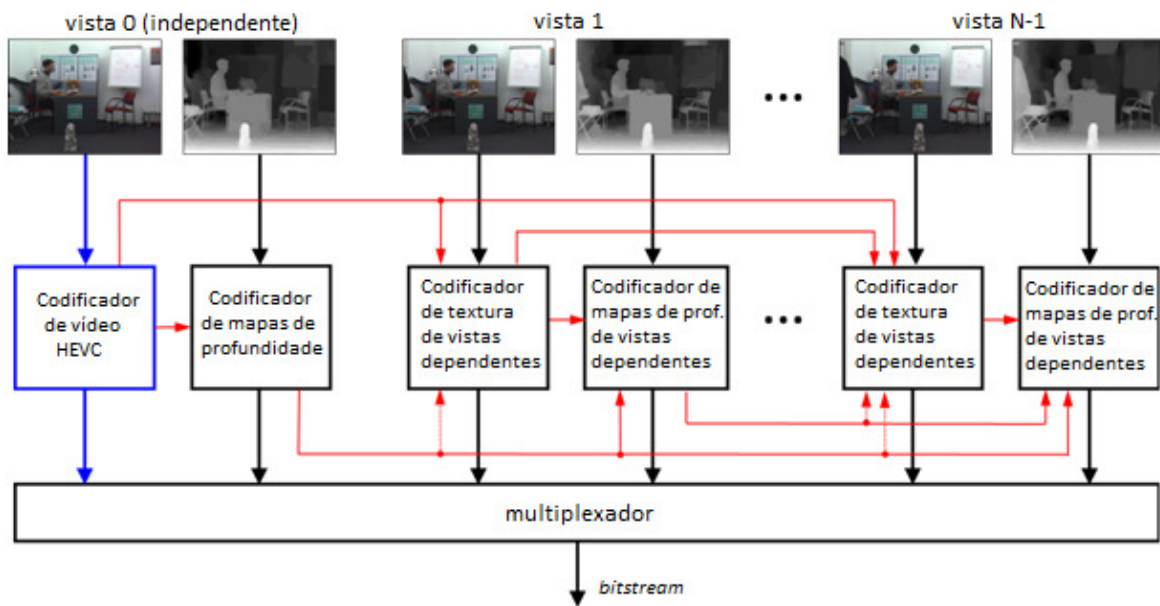


Figura 3.3: Estrutura básica do codificador 3D-HEVC [11].

Algumas das principais ferramentas acrescentadas ao 3D-HEVC para a codificação das vistas dependentes e mapas de profundidade, as quais estendem o HEVC são: predição com compensação de disparidade; predição inter-vista do movimento; predição inter-vista dos resíduos; codificação dos mapas de profundidade utilizando novos modos de codificação intra; herança de parâmetros de movimento; e predição das *quadrees* de profundidade. Tais ferramentas serão descritas em mais detalhe nas subseções a seguir.

3.2.3. Ferramentas de Codificação

A codificação de vídeo multivista tira partido da grande quantidade de informações redundantes entre as vistas pois, tendo em conta que as câmeras capturam a mesma cena a partir de diferentes pontos de vista no mesmo instante de tempo, o conteúdo de cada vista é usualmente bastante semelhante ao das vistas espacialmente próximas. Essas redundâncias podem ser exploradas utilizando predição inter-vistas, tal como representado na Figura 3.3 onde se indicam as relações de predição entre vistas e entre componentes. Desta forma as tramas podem ser preditas não somente a partir de tramas vizinhas temporalmente, mas também a partir de tramas já codificadas em vistas vizinhas. A seguir serão apresentadas algumas das principais ferramentas acrescentadas ao codec HEVC, as quais utilizam informações de vistas previamente codificadas para otimizar o processo de codificação das vistas dependentes no 3D-HEVC.

3.2.3.1. Predição com Compensação de Disparidade

Para compartilhar a informação de textura previamente codificada nas vistas adjacentes, a Predição com Compensação de Disparidade (*Disparity-Compensated Prediction - DCP*) foi adicionada como uma alternativa à Predição com Compensação de Movimento (*Motion-Compensated Prediction - MCP*). De modo que, a MCP refere-se a predição de inter-trama, a qual utiliza como referência as tramas já codificadas da mesma vista em instantes de tempo diferentes, enquanto a DCP refere-se a predição inter-vista, a qual utiliza tramas já codificadas em outras vistas, mas no mesmo instante de tempo [12], conforme ilustrado na Figura 3.4. Nesta figura, o índice da trama de referência chamado R, indica se a predição dos blocos inter-codificados da trama atual foi realizada utilizando MCP, como ocorre com os blocos que usam as tramas com índice R igual 0, 2 e 4; ou utilizando a DCP, como ocorre com o bloco em amarelo que utiliza a trama com índice R igual a 1.

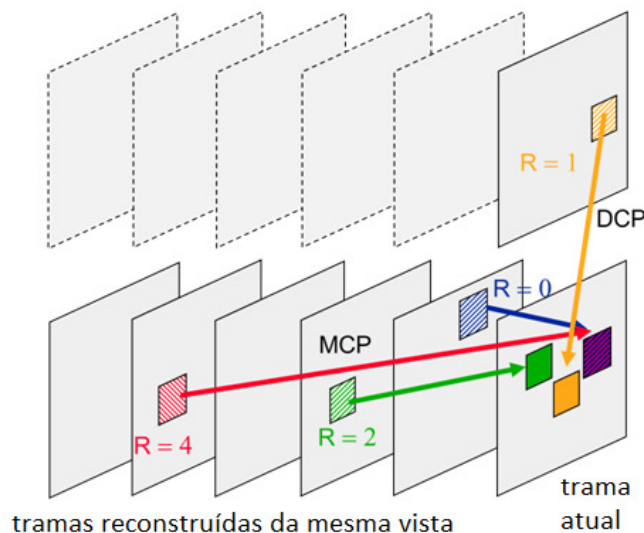


Figura 3.4: Exemplo de DCP e MCP para codificação de vistas dependentes [11].

3.2.3.2. Predição Inter-Vistas de Movimento e Resíduo

Uma vez que as vistas de uma sequência de vídeo representam diferentes projeções da mesma cena 3D capturadas através de múltiplas câmeras, o movimento nas diferentes vistas é muito similar, o que faz com que os parâmetros de movimento para codificar uma vista dependente possam ser preditos a partir dos parâmetros de movimento de uma vista já codificada, conforme ilustrado na Figura 3.5. Nesta figura, para calcular ou estimar um vetor de movimento para um bloco da vista atual, um vetor de disparidade é usado para localizar a posição do bloco co-localizado na trama já codificada da vista de

referência. Se esse bloco foi codificado usando MCP, os seus parâmetros de movimento (vetores de movimento e índices de referência) podem ser usados como candidatos para codificação do bloco da vista atual. Além disso, o vetor de disparidade derivado pode ser também diretamente usado como um vetor candidato para a DCP [60][61].

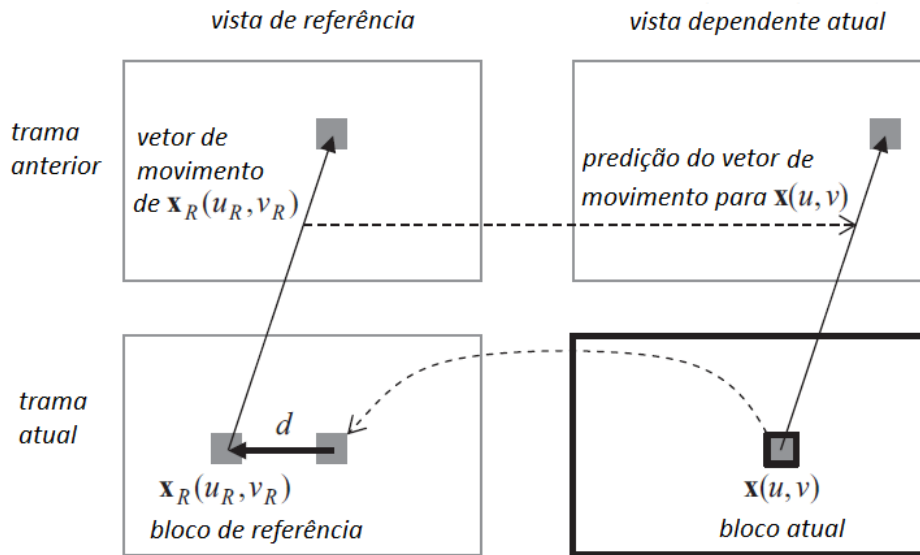


Figura 3.5: Vetor de movimento predito a partir do bloco correspondente da vista de referência já codificada [12][44].

Além da informação de movimento predita a partir das vistas adjacentes, é possível também realizar a predição inter-vista dos dados de resíduos (de predição temporal) usando um processo semelhante ao descrito acima. De modo que após encontrar o bloco co-localizado na vista de referência usando o vetor de disparidade associado, o sinal residual temporal reconstruído desse bloco é utilizado na predição dos resíduos temporais do bloco da vista atual. Assim, as amostras residuais do bloco de referência são subtraídas das amostras do bloco atual e somente a diferença é usada na codificação com transformada e quantização [12].

3.2.3.3. Compensação de Iluminação (IC)

Quando diferentes câmeras que capturam a mesma cena não estão calibradas (na cor ou luminância), a predição inter-vistas pode falhar ou ser pouco eficiente [62].

A compensação de iluminação foi introduzida na codificação multivista para melhorar a eficiência de codificação de blocos preditos a partir de tramas de outras vistas, utilizando predição inter-vista. Na Figura 3.6, são apresentadas tramas de vídeo capturadas a partir de diferentes câmeras, com iluminação visivelmente distinta. Em casos como esse,

a compensação de iluminação é utilizada para diminuir a diferença de pixels no mesmo objeto de cada vista, conforme explicado a seguir.



Figura 3.6: Variação de iluminação e crominância em tramas capturadas por câmeras diferentes [63].

Para adaptar as amostras de luminância e crominância dos blocos preditos inter-vista à iluminação da vista atual, um modelo de compensação de iluminação linear é utilizado no 3D-HEVC. Tal modelo tem seus parâmetros estimados para cada CU utilizando as amostras vizinhas reconstruídas do bloco da vista atual e do bloco de referência usado na predição inter-vista, conforme ilustrado na Figura 3.7.

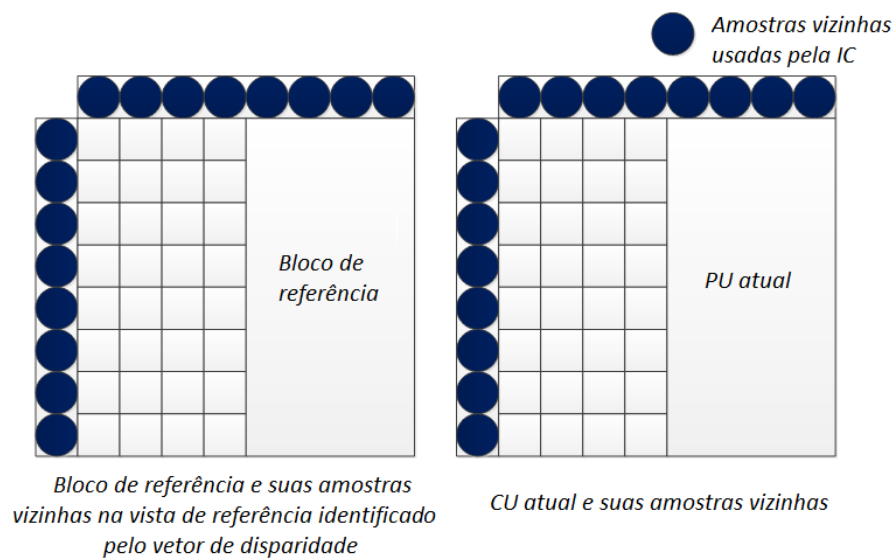


Figura 3.7: Amostras vizinhas usadas no processo de compensação de iluminação [11].

A compensação de iluminação só se aplica aos blocos que são preditos usando predição inter-vista. Além disso, tal compensação só é aplicada para a codificação de blocos com um modo de partição de $2N \times 2N$ [11].

3.2.4. Codificação de Mapas de Profundidade

Durante a codificação dos mapas de profundidade, os mesmos conceitos de predição com compensação de disparidade e predição com compensação de movimento usados na codificação das vistas de textura, são utilizados. No entanto, conforme mencionado anteriormente, os mapas de profundidade são caracterizados por possuírem arestas bem definidas e grandes regiões homogêneas [61]. Deste modo, novas técnicas de predição intra usando Modos de Modelagem da Profundidade, chamados de DMMs (*Depth Modelling Modes*), foram desenvolvidas para codificar de forma mais eficiente as arestas bem definidas. Além disso, tendo em conta que os filtros de interpolação de 8 *taps* usados na compensação de movimento das vistas de textura podem produzir artefatos nas bordas sobressaídas dos mapas de profundidade, que podem ser visíveis nas vista intermédias sintetizadas, os processos de MCP e DCP foram modificados de forma que nenhuma interpolação é utilizada nesses métodos durante a codificação da profundidade. Assim, a predição inter-tramas dos mapas de profundidade é sempre realizada com precisão de amostra completa (*full-sample*) [11].

Ao contrário do que sucede com as vistas do vídeo cuja qualidade avaliada diretamente a partir da sua versão descodificada, a qualidade dos mapas de profundidade (após a sua codificação) tem que ser avaliada indiretamente, através da qualidade das vistas sintetizadas [12].

3.2.4.1. Modos de Modelagem da Profundidade (DMMs)

Conforme dito anteriormente, os mapas de profundidade possuem duas características principais: arestas bem definidas, as quais representam os limites dos objetos, e grandes áreas homogêneas com valores de amostra quase constantes, as quais geralmente representam as áreas planas dos objetos e o plano de fundo da imagem orientados paralelamente ao plano do sensor da câmara, conforme ilustrado na Figura 3.8.

Dado que a predição intra e codificação com transformadas utilizadas no HEVC podem gerar artefatos significativos durante a codificação das bordas bem definidas dos mapas de profundidade, causando distúrbios visíveis ao longo dos limites dos objetos nas vistas sintetizadas, uma melhor representação dessas bordas foi incluída no 3D-HEVC

através da adição de novos modos de predição intra ao processo de codificação de profundidade, os chamados DMMs.

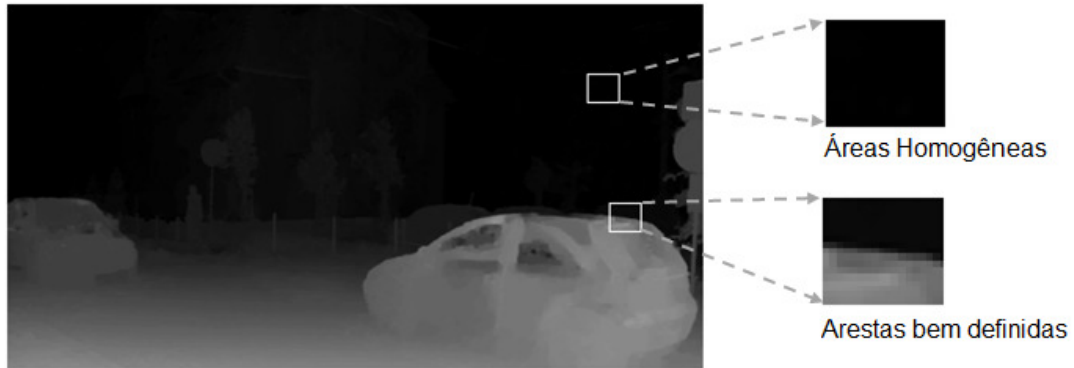


Figura 3.8: Exemplo de mapa de profundidade da sequência PoznanStreet, destacando as características de bordas bem definidas e áreas homogêneas.

Assim, alternativamente aos modos de predição intra convencionais usados no HEVC, os DMMs foram incluídos na codificação intra visando uma melhoria na eficiência de codificação das bordas sobressaídas dos mapas de profundidade [11].

Para a utilização desses modos, a PU de profundidade é dividida em duas regiões não retangulares, representadas por P1 e P2 na Figura 3.9 (a-b), em que cada região é aproximada por um valor constante, comumente chamado de Valor Constante de Partição (CPV).

Dois parâmetros são necessários para definir o bloco de profundidade, isto é, a informação de particionamento, que indica a que região cada pixel pertence, e o CPV, indicando um valor constante para as amostras da região correspondente [12].

Em um dos casos mais simples de codificação de uma PU de profundidade usando os DMMs, a PU é dividida em duas regiões delimitadas por uma linha reta representando uma aproximação da borda de profundidade, este método de especificação de partição é normalmente designado como partição Wedgelet, ilustrado na Figura 3.9 (a).

Quando o bloco de profundidade apresenta um padrão mais complexo, o correspondente bloco de textura é utilizado para obter o contorno do bloco de profundidade. Este método de particionamento é ilustrado na Figura 3.9 (b) é chamado Partição do Contorno.

A Figura 3.9 (c) ilustra uma partição Wedgelet e uma Partição de Contorno para dois blocos de profundidade tomados a partir de uma trama do vídeo UndoDancer.

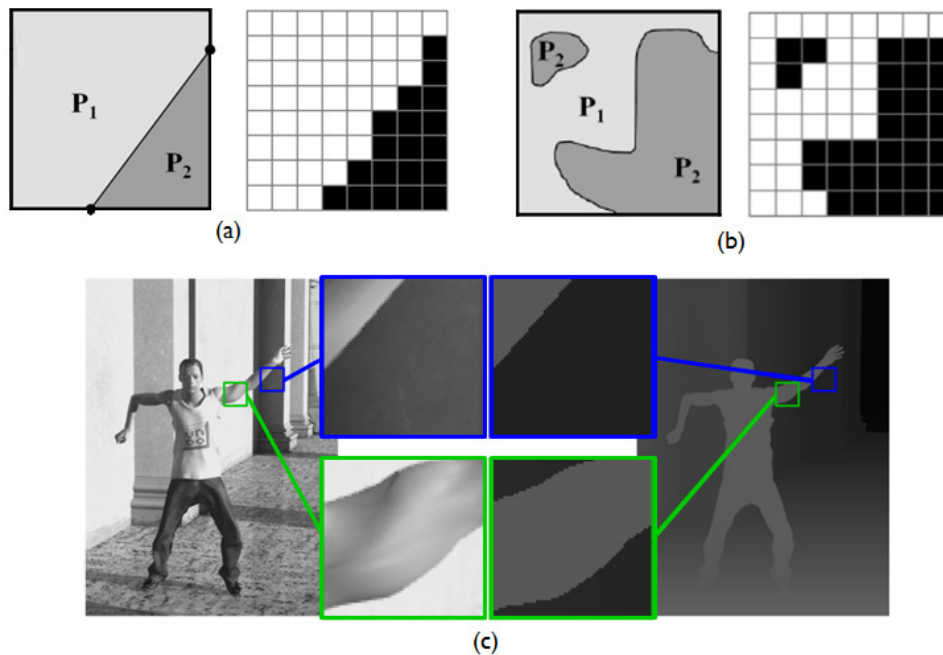


Figura 3.9: Partição Wedgelet (a) e Partição de Contorno (b): Espaço de sinal contínuo (à esquerda) e padrão de partição correspondente (à direita). Trama de profundidade da seqüência UndoDancer (c) exemplo de informação de Partição Wedgelet (azul) e Partição de Contorno (verde) [61].

Assim, dois tipos diferentes de partições são utilizados: baseados em Wedgelets e em Contornos, os quais diferem pela maneira como a segmentação do bloco de profundidade é realizada [12].

Nas versões iniciais do software de referência do 3D-HEVC, quatro DMMs eram incluídos na codificação intra [61]. No entanto, tendo em vista que os ganhos em eficiência de codificação de alguns modos eram muito baixos em relação a complexidade computacional acrescida quando eram utilizados, na versão atual do software de referência 3D-HEVC [11] somente dois DMMs permaneceram: o modo de Sinalização Wedgelet Explícita, chamado de DMM1, e o modo de Predição inter-componente das Partições de Contorno, chamado de DMM4, os quais serão apresentados em mais detalhe nas seções a seguir.

3.2.4.1.1. Sinalização Explícita da Wedgelet

Neste modo de modelagem de profundidade, a partição Wedgelet com melhor correspondência é estimada no codificador e transmitida no bitstream. No codificador, uma pesquisa considerando um conjunto de partições Wedgelet é realizada utilizando o sinal de profundidade original do bloco atual como uma referência. Durante esta pesquisa, a

partição Wedgelet que produzir a menor distorção entre o sinal original e a aproximação Wedgelet é selecionada. No decodificador, o bloco de amostras de profundidade é reconstruído utilizando a informação de partição transmitida, de modo que a informação de partição wedgelet para este modo não é predita [12]. O sinal de predição resultante é, então, avaliado utilizando o processo de decisão de modo tradicional.

Cada partição Wedgelet contém uma posição de partida (S na Figura 3.10) e outra de final (E na Figura 3.10), definidas entre os pixels que fazem parte dos limites do bloco que está a ser codificado. Visando reduzir o número de partições Wedgelet, algumas restrições sobre as posições de ponto de partida e final de partições Wedgelet têm sido aplicadas, sendo as Wedgelets definidas por uma de seis direções e classificadas em duas categorias: partições de borda adjacentes e partições de borda opostas.

As partições de borda adjacentes têm posição inicial e final pertencentes as linhas de fronteira verticais e horizontais adjacentes da PU atual, conforme ilustrado na Figura 3.10 com $wedgeOri$ variando de 0 a 3, enquanto partições de borda opostas têm posição inicial e final pertencentes a linhas de fronteira paralelas da PU atual, como indicado na Figura 3.10 com $wedgeOri$ igual a 4 ou 5.

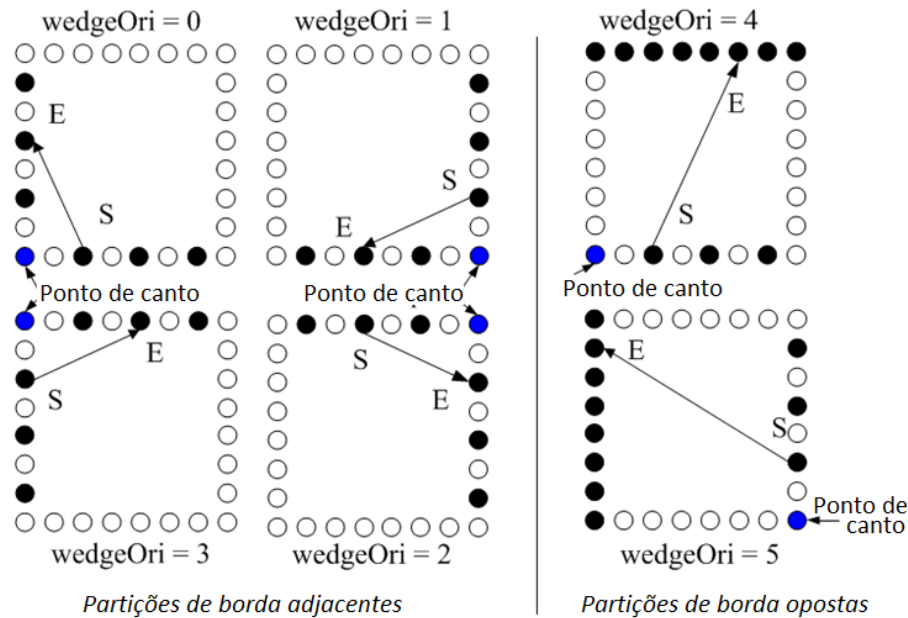


Figura 3.10: Seleção das Partições Wedgelet [11].

Para uma partição de borda adjacente, um ponto de canto da partição Wedgelet é definido como uma amostra que se encontra entre os quatro cantos do bloco atual e tem a distância mais próxima até às posições inicial e final. Além disso, as posições inicial e final

devem ser pares relativamente ao canto correspondente, como pode ser visto na Figura 3.10.

Para as partições de borda opostas, o ponto de canto é definido como o canto inferior esquerdo para *wedgeOri* igual a 4, e canto inferior direito da PU para *wedgeOri* igual a 5, conforme ilustrado na Figura 3.10. Além disso, para as partições de borda opostas, a sua posição inicial deve ter um número par de distância até o ponto de canto.

Na Figura 3.10, as posições de pixel representados com pontos cheios, podem ser escolhidas como posições de início, indicadas como S na Figura 3.10, ou final, indicadas como E, para cada tipo de partição Wedgelet, com *wedgeOri* que variam de 0 a 5.

3.2.4.1.2. Predição Inter-componente das Partições de Contorno

Este modo tem como objetivo prever uma partição de Contorno a partir de um bloco de referência de textura através de predição inter-componente. Tal como para a predição inter-componente de um padrão de partição Wedgelet, o sinal de luminância reconstruído do bloco co-localizado da trama de vídeo associada é usado como referência. Diferente das partições Wedgelet, um método baseado na comparação com um limiar é usado para a predição de uma partição de contorno. Neste método, o valor médio das quatro amostras de luminância dos cantos do bloco de referência de textura é definido como limiar *e*, dependendo se o valor de uma amostra está acima ou abaixo desse limiar, a posição da amostra é categorizada como parte da região P1 ou P2 no padrão da partição de contorno resultante [61].

3.2.4.2. Herança dos Parâmetros de Movimento

Tendo em vista que as características do movimento da textura do vídeo e de seus mapas de profundidade associados são semelhantes, pois ambos são projeções da mesma cena a partir do mesmo ponto de vista e no mesmo instante de tempo, um modo de herança dos parâmetros de movimento, chamado MPI (*Motion Parameter Inheritance*) foi desenvolvido a fim de codificar de forma mais eficiente os dados dos mapas de profundidade.

Ao utilizar esse modo, o particionamento de uma PU de profundidade em sub-PUs e os parâmetros de movimento associados são inferidos a partir da PU co-localizada na trama de textura associada.

Cada PU dentro da trama de profundidade atual, é inicialmente dividida em várias sub-PU's $N \times N$ e logo após uma predição de movimento inter-vista das sub-PU's é aplicada, de forma que diferentes vetores de movimento podem ser herdados para sub-PU's $N \times N$ de uma mesma PU de profundidade.

Uma vez que os vetores de movimento do sinal de vídeo têm precisão de quarto de amostra, e o sinal do mapa de profundidade utiliza precisão de amostra inteira (*full-sample*), os vetores de movimento herdados são quantizados para precisão de posição inteira [64].

Para cada PU de profundidade, pode ser decidido de forma adaptativa se as informações de particionamento e movimento serão herdadas a partir da PU co-localizada na trama de textura do vídeo, ou se novos dados de movimento serão transmitidos.

3.2.4.3. Predição das Quadrees de Profundidade

A predição das *quadrees* de profundidade é realizada a partir das *quadrees* utilizadas na textura, de modo que a profundidade das *quadrees* da profundidade é limitada pelo grau de particionamento utilizado na textura. Assim, cada CTU da trama de profundidade está diretamente relacionada com a sua CTU co-localizada na trama de textura, e uma determinada CU da profundidade não pode ter mais divisões do que a CU co-localizada na textura [61].

A Figura 3.11 apresenta os tipos de particionamento possíveis das *quadrees* de profundidade em relação as partições de textura. A partir da figura é possível observar que quando uma CU de textura utiliza uma partição $N \times N$, a CU de profundidade pode utilizar todos os tipos de particionamento possíveis até $N \times N$.

No entanto, quando a partição da CU de textura é $2N \times 2N$, $N \times 2N$, $2N \times N$ e $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$ ou $nR \times 2N$ [11], a CU de profundidade pode utilizar somente partições $2N \times 2N$.

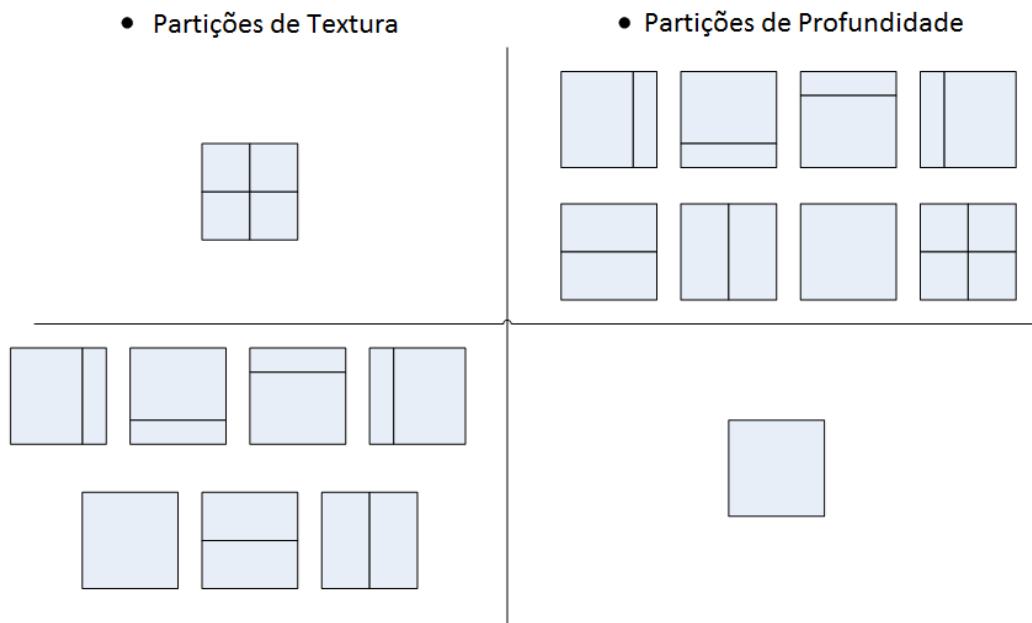


Figura 3.11: Partições de profundidade possíveis a partir das partições de textura utilizadas [11].

Neste capítulo foram apresentadas algumas das aplicações mais recentes de vídeo 3D, bem como uma descrição da estrutura e principais técnicas de codificação adicionadas à extensão 3D-HEVC para a codificação das vistas de textura e mapas de profundidade.

Os Capítulos 4 e 5 que se seguem apresentam os algoritmos desenvolvidos para concretizar as contribuições originais desta dissertação descritas abreviadamente na Seção 1.3.

4. ALGORITMOS RÁPIDOS PARA A CODIFICAÇÃO INTRA DO HEVC

Recentemente, algumas comparações foram realizadas entre os padrões H.264/AVC e HEVC. Os resultados obtidos demonstraram que o HEVC pode alcançar uma redução na taxa de bits de aproximadamente 50% em média, quando comparado ao H.264/AVC, especialmente para conteúdos de vídeo de alta definição [5][65][66]. No entanto, para a configuração *all-Intra* (AI) foi observada somente uma redução de aproximadamente 21,9% na taxa de bits, com um aumento de complexidade semelhante ao da configuração *random access* (RA) [67]. Além disso, a decisão do modo de predição intra é responsável por mais da metade da complexidade computacional total da codificação intra-trama segundo testes efetuados com o software de referência do HEVC (*HEVC Test Model*) [42][68].

Na configuração AI todas as tramas são codificadas usando somente predição intra, de modo que tramas de referência temporal não são usadas [69][4]. Uma vez que a configuração AI evita o alto custo do processo de estimação de movimento e requer menos memória do que a codificação inter-tramas, a sua utilização é importante principalmente quando necessita-se de um codec simples e que demande menos energia.

A configuração AI é interessante para utilização em dispositivos de baixa complexidade, como câmeras de vídeo sem fio, telefones portáteis e PDAs, os quais geralmente possuem menos poder de processamento ou suprimento de energia limitado [70][71].

A codificação AI também é útil em configurações de vídeo vigilância, câmeras portáteis com qualidade de estúdio e redes de distribuição de cinema digital [72][73], uma vez que ela permite uma edição simples e rápida, sem a perda de qualidade que ocorre durante a edição de um vídeo codificado com recurso a predição temporal. Portanto, a investigação relacionada à aceleração da codificação *all-intra* (AI) do HEVC é relevante para muitas áreas de aplicação distintas.

As principais contribuições desta tese estão focadas na otimização da codificação HEVC e extensão 3D-HEVC utilizando a configuração *all-intra* (AI), como poderá ser visto no decorrer das próximas seções.

4.1. Otimização da Predição Intra

Conforme descrito no Capítulo 2, o padrão HEVC define que cada trama é dividida em CTUs, as quais podem ser particionadas em unidades de codificação (CUs) através de uma estrutura de árvore recursiva. As CUs-folha podem ainda ser particionadas em unidades de predição chamadas de PUs [5]. De modo que para predição intra, há dois tipos de particionamento de PUs possíveis: uma PU de tamanho $2N \times 2N$ ou quatro PUs de tamanho $N \times N$, onde N corresponde a metade do tamanho da CU, conforme ilustrado na Figura 4.1.

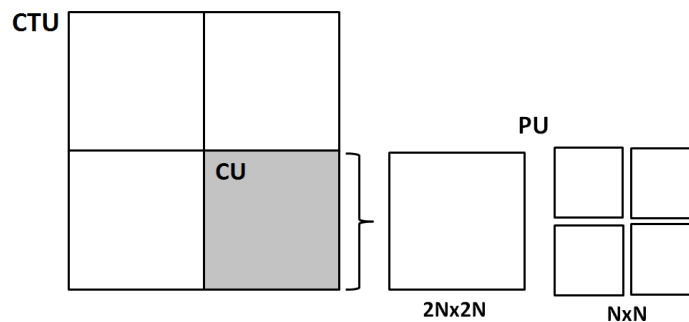


Figura 4.1: Tipos de particionamento de PUs intra.

O processo de predição intra utilizado no HEVC é muito semelhante ao usado no H.264/AVC, no qual as amostras do bloco a ser codificado são preditas direcionalmente utilizando como referência as amostras da periferia dos blocos ou PUs espacialmente vizinhos à esquerda e acima. No entanto, diferentemente do H.264/AVC, o qual utiliza somente oito direções de predição, a nova estrutura de predição definida no HEVC admite um número aproximadamente quatro vezes maior de modos intra, apresentando um total de 33 direções de predição, além dos modos DC e Planar, conforme apresentado na Figura 4.2.

Este aumento no conjunto de direções de predição resulta em um custo computacional muito maior para todo o processo de codificação intra. Além disso, este custo em complexidade é ainda ampliado pelo uso de estruturas de particionamento recursivo em *quadtree*, pois o codificador deve definir ambos a profundidade da árvore e a direção de predição a ser utilizada na codificação de cada CU da *quadtree*.

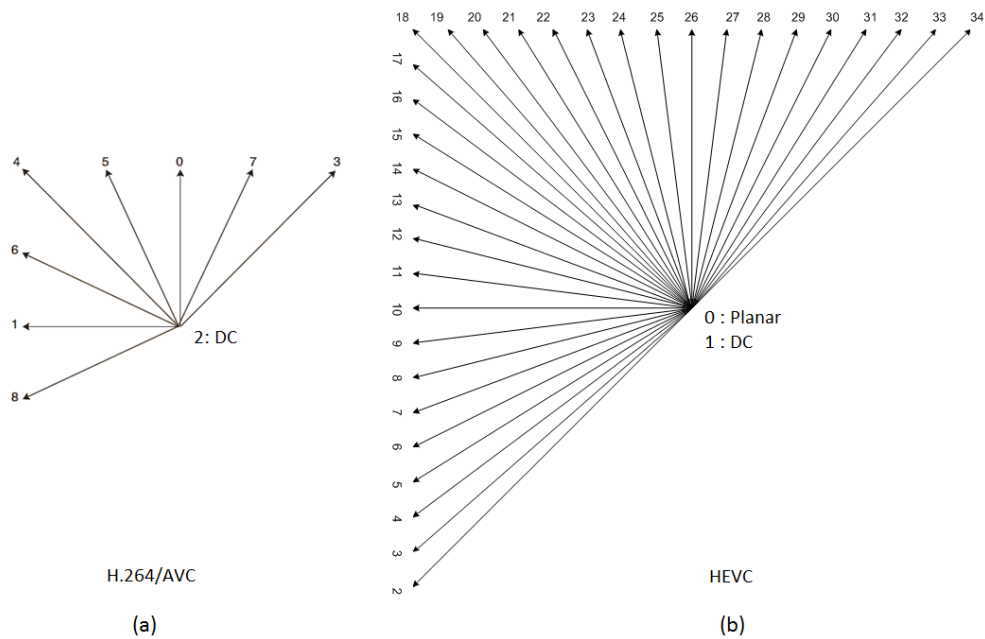


Figura 4.2: Direções de Predição Intra dos padrões H.264/AVC (a) e HEVC (b).

Para identificar os componentes que consomem mais tempo durante a codificação, a complexidade computacional do codificador HM (software de referência do HEVC) foi recentemente analisada (*profiled*) em [19]. Os resultados demonstraram que, no caso da configuração AI, cerca de um quarto do tempo total de codificação é gasto no processo de RDOQ (*Rate-Distortion Optimized Quantization*) e 16% do tempo é gasto na predição intra [19], conforme apresentado na Figura 4.3. Para reduzir a complexidade computacional da predição intra, um processo de decisão de modo simplificado (RMD) foi incluído no HM desde a versão HM 2.0 [74], visando diminuir o número de modos intra a serem avaliados durante o estágio de RDOQ.

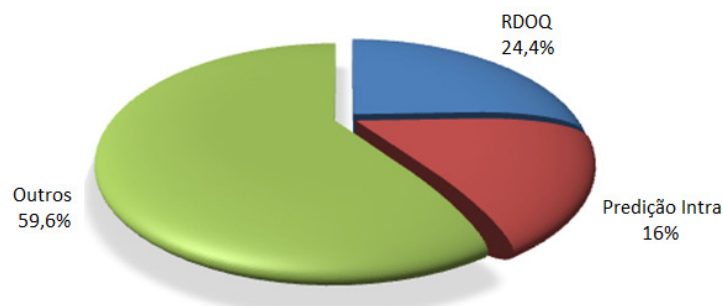


Figura 4.3: Distribuição da complexidade de codificação usando *all-intra* no codificador HM.

Neste processo, todos os 35 modos disponíveis são testados para selecionar os modos que irão compor um subconjunto de modos de predição candidatos. Tal

subconjunto é composto pelos modos que produzem a menor soma das diferenças absolutas dos coeficientes transformados (SATD - *Hadamard Transformed SAD*) entre os pixels originais e os pixels preditos. No cálculo de SATD, uma subtração do bloco original pelo bloco predito é realizada e, então, é aplicada a transformada *Hadamard* aos resultados da subtração [5] [17], conforme apresentado na Figura 4.4. Os valores absolutos resultantes da transformada *Hadamard* são somados para gerar a distorção do bloco predito em relação ao bloco original. Desta forma, os modos intra que apresentarem menor valor de SATD (ou seja, menor distorção) são selecionados para fazerem parte do subconjunto de modos candidatos.

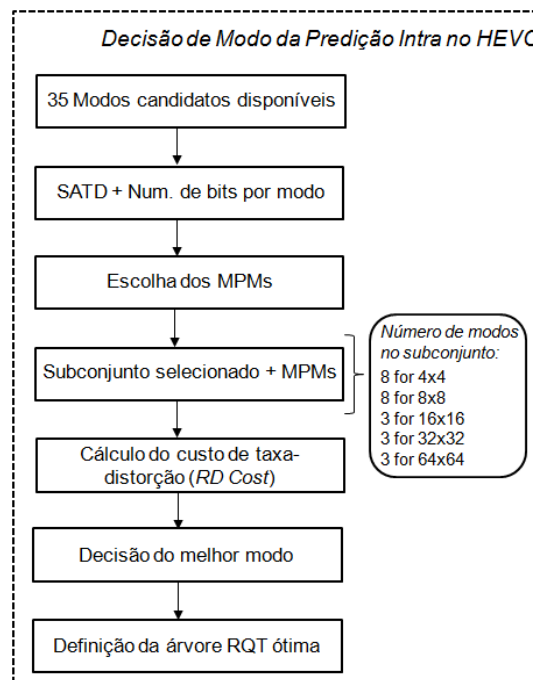


Figura 4.4: Decisão de modo na predição intra do HEVC.

Além disso, a este subconjunto são adicionados os modos mais prováveis (MPMs - *Most Probable Modes*), conforme ilustrado na Figura 4.4. Em contraste com o padrão H.264/AVC, o qual utilizava apenas um MPM, o HEVC inclui três MPMs devido ao aumento no número de direções de predição intra [5].

Os três MPMs são escolhidos com base nos modos intra selecionados nas PUs vizinhas à PU que está a ser codificada. Normalmente, os dois primeiros MPMs correspondem aos modos das PUs vizinhas à esquerda e acima, e o terceiro MPM é definido como sendo o modo Planar, o modo DC, ou o modo angular 26 (nesta ordem de prioridade) considerando a atribuição dos dois primeiros MPMs, de modo a não duplicá-

los. Além disso, nos casos em que as PUs vizinhas (à esquerda e acima) possuem o mesmo modo intra, este modo e os dois modos direcionais (angulares) mais próximos são escolhidos para compor o conjunto de três MPMs [20].

O tamanho deste subconjunto depende do tamanho da PU. Para PUs 4x4 e 8x8, o subconjunto inclui os oito modos com o melhor desempenho, enquanto que para PUs com tamanho 16x16, 32x32 e 64x64 o subconjunto inclui apenas os três modos com os melhores resultados.

Logo após a definição do subconjunto de modos candidatos, o custo RD (*Rate-Distortion*) de cada modo de predição pertencente a este subconjunto é calculado e, a partir deste cálculo, o modo com o menor custo é selecionado como melhor modo para codificar a PU.

O RD *Cost* é calculado de acordo com a função de custo J [75] definida em (4.1) [17], na qual D_{SATD} corresponde à distorção calculada através de SATD, R_{modo} representa o número de bits para cada modo de predição, e λ é o multiplicador de Lagrange que é calculado em função do passo de quantização $\lambda(QP)$.

$$J = D_{SATD} + \lambda(QP) \cdot R_{modo} \quad (4.1)$$

Por fim, a definição da árvore residual (*Residual Quadtree - RQT*) ótima a ser usada para codificar o resíduo obtido utilizando o modo de predição intra selecionado conclui o processo. É importante ressaltar que, todas as configurações possíveis de árvore residual, com todos os tamanhos de TU possíveis, são testados no processo RDO.

Considerando o processo RMD exposto acima, uma análise específica do procedimento de predição intra do HM foi realizada para identificar as operações que mais tempo consomem neste processo [76]. Tal análise considerou os três principais módulos que compõem a predição intra do HM, avaliando a complexidade de cada um em relação à complexidade total da predição intra. O primeiro módulo avaliado foi a “Seleção de N Candidatos”, o qual se baseia no custo SATD e seleciona os N modos com menor custo. O segundo módulo foi o “RD Simplificado”. Neste módulo, o custo RD dos N modos selecionados na etapa anterior são avaliados utilizando uma TU de tamanho fixo com o mesmo tamanho da PU que está a ser codificada. Finalmente, o terceiro módulo chamado “RD completo” é calculado usando o modo intra escolhido na etapa anterior. Este módulo utiliza tamanhos de TUs menores a partir do tamanho da PU que está sendo codificada

para encontrar o melhor tamanho de TU. O cálculo de "RD completo" é aplicado apenas uma vez para cada PU [76].

Conforme ilustrado na Figura 4.5, os resultados desta análise de complexidade mostram que as percentagens de tempo de processamento gasto em cada um destes módulos são de 19%, 56% e 24%, respectivamente. Uma vez que o segundo módulo é executado tantas vezes quanto o número de modos de candidatos, este módulo é o que mais consome tempo no processo (56%). Um parâmetro chave deste módulo é o número de modos candidatos selecionado no primeiro módulo, de forma que quanto maior o número de candidatos mais complexo é o processo de codificação intra.

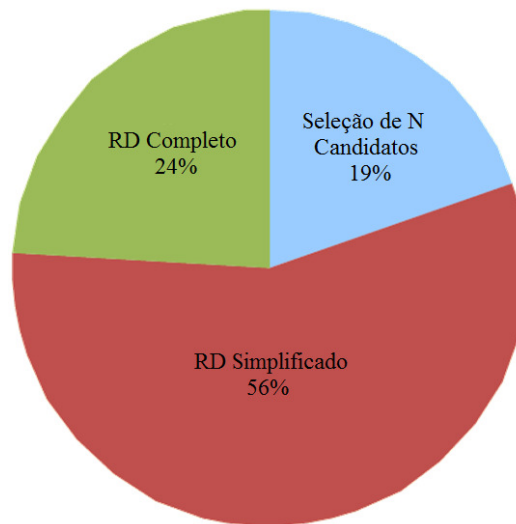


Figura 4.5: Análise de complexidade dos três principais módulos da predição intra do HM [76].

Os algoritmos propostos neste capítulo reduzem significativamente o tempo consumido para a decisão do modo de predição intra, otimizando o primeiro e segundo módulo mencionados acima, conforme apresentado em detalhe nas próximas seções.

Apesar do processo de decisão do modo de predição intra do HEVC utilizar a versão simplificada RMD (*Rough Mode Decision*), este processo ainda é muito mais complexo do que aquele utilizado no H.264/AVC [77]. Este fato e a importância da predição intra em aplicações específicas de codificação de vídeo, tais como as citadas no início desse capítulo, fazem com que a procura de soluções para reduzir a complexidade da predição intra do HEVC seja uma importante questão de pesquisa.

Alguns trabalhos descrevendo métodos que visam reduzir a carga computacional do processo de predição intra do HEVC têm sido publicados. No trabalho publicado por Zhao et al. [78] um processo de decisão de modo simplificado (RMD) é

realizado para reduzir o número de modos intra candidatos a serem avaliados durante o processo de otimização da taxa de distorção (RDO). Este trabalho também explora a forte correlação entre os modos de blocos vizinhos, para identificar o modo mais provável (MPM), o qual é adicionado ao subconjunto de modos candidatos.

No trabalho publicado por Kim et al. [79], um algoritmo rápido para decisão do tamanho da CU a ser usada na codificação intra é proposto, visando terminar o processo de divisão da CU com antecedência, antes da verificação do custo RD da divisão em sub-CUs.

No algoritmos proposto por Jiang et al. [80] um histograma de direções de gradiente é calculado para cada CU, e com base neste histograma um pequeno subconjunto dos modos candidatos é selecionado como entrada para o processo de RDO.

O método proposto por Tian e Goto [81] apresenta um algoritmo de decisão do tamanho da PU para acelerar a codificação intra do HEVC, o qual é realizado em duas etapas. Na primeira etapa, antes do início do processo de predição intra, a complexidade textura de cada CTU e seus sub-blocos são avaliados para filtrar as PUs que não serão usadas, assim, se a complexidade da CTU é considerado alta, as PUs maiores são excluídas do processamento; se a complexidade é considerada baixa, as PUs menores (8×8 e 4×4) são filtradas. Na segunda etapa, durante o processo de codificação intra, os tamanhos de PU vizinhas já codificadas são utilizados para decidir se PUs menores devem ser testadas ou não na codificação da PU atual.

No método proposto por Kim et al. [82] uma terminação antecipada da predição intra é realizada com base no modo de predição intra utilizado na PU da profundidade anterior e no tamanho de TU da profundidade atual. Além disso, o número de candidatos do processo RMD é também reduzido antes do cálculo de RDO, e o modo intra da PU da profundidade anterior é sempre incluído na lista de candidatos para decisão de modo intra.

No trabalho publicado por Zhang et al. [83], um modo de decisão rápido adaptativo é proposto para a predição intra do HEVC, o qual analisa as características de textura das PUs para reduzir o número de modos de candidatos a serem avaliados no processo RDO. No método proposto por Zhang e Ma [84], uma sub-amostragem espacial 2:1 é aplicada à predição residual e, em seguida, uma transformada Hadamard é aplicada sobre o resíduo da predição sub-amostrada para derivar o SATD a ser utilizado no processo de decisão RMD. Além disso, uma pesquisa gradual para reduzir o número de modos para

o cálculo de custo da Hadamard e um esquema de terminação antecipada é aplicado para acelerar o processo de RDO.

Os algoritmos rápidos para a predição intra do HEVC propostos neste capítulo levam em consideração três correlações principais: (1) a correlação entre o modo de predição intra e a orientação de borda dominante da textura da PU; (2) a correlação entre o modo de predição intra da PU atual e os modos das PUs de níveis anteriores de profundidade da árvore; e (3) a correlação entre os conjuntos de modos selecionados.

Nas próximas seções serão apresentados em detalhe os algoritmos propostos para redução da complexidade computacional da predição intra e, conseqüentemente, da complexidade global do codificador HEVC.

4.1.1. Orientação da Borda Dominante

Conforme mencionado anteriormente na Seção 4.1, desde a versão 2.0 do HM a decisão do modo de codificação intra utilizado no HEVC inclui um processo de decisão simplificado (RMD), o qual seleciona um subconjunto de modos candidatos para ser utilizado na predição de cada PU. No entanto, esta solução deve avaliar com antecedência todos os 35 modos de predição intra definidos no padrão HEVC, a fim de selecionar os modos que compõem este subconjunto. Uma vez que a avaliação de cada um dos modos envolve uma série de cálculos complexos, a complexidade computacional da decisão de modo ainda é muito grande, e o tempo de codificação ainda é bastante significativo.

No algoritmo proposto nesta subseção, ao invés de utilizar o conjunto completo de 33 modos angulares, mais os modos DC e Planar, no processo de decisão do modo intra, um subconjunto menor de modos é selecionado de acordo com a orientação de borda dominante calculada a partir da textura da PU.

Inicialmente, a textura das PUs é analisada para determinar a sua orientação de borda dominante, a qual é utilizada para selecionar o subconjunto de modos que serão avaliados no procedimento de codificação intra. A orientação dominante permite limitar os modos a pesquisar àqueles cuja direção de predição é próxima dessa orientação. Cinco indicadores de orientação de borda são avaliados: horizontal, vertical, diagonal 45°, diagonal 135° e não-direcional [85].

A orientação de borda dominante de cada PU 4x4 é calculada com base nos valores de luminância dos seus pixels. Cada PU 4x4 é dividida em quatro blocos 2x2: *Bloco 0*, *Bloco 1*, *Bloco 2* e *Bloco 3*, conforme ilustrado na Figura 4.6 e, logo após, a média

das amostras de luminância de cada bloco é calculada. Estas médias são denotadas como c_0 , c_1 , c_2 e c_3 , respectivamente.

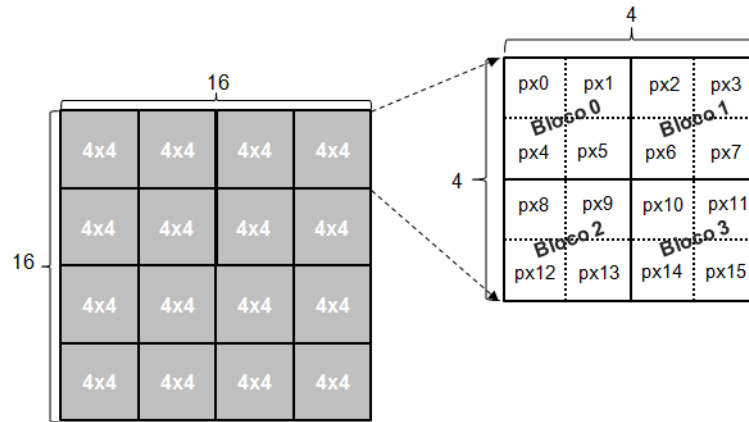


Figura 4.6: Cálculo de borda para uma PU 16x16.

Estes valores médios são então utilizados nas equações de (4.2) a (4.6) para calcular os cinco indicadores de orientação (ou força de borda) [85]: eV , eH , $e45^\circ$, $e135^\circ$ e eND (não-direcional)

$$eV = |c_0 - c_1 + c_2 - c_3| \quad (4.2)$$

$$eH = |c_0 + c_1 - c_2 - c_3| \quad (4.3)$$

$$e45^\circ = |\sqrt{2} \times (c_0 - c_3)| \quad (4.4)$$

$$e135^\circ = |\sqrt{2} \times (c_1 - c_2)| \quad (4.5)$$

$$eND = |2 \times (c_0 - c_1 - c_2 + c_3)| \quad (4.6)$$

onde eV significa borda Vertical, eH significa borda Horizontal, $e45^\circ$ corresponde à borda Diagonal 45° , $e135^\circ$ corresponde a borda Diagonal 135° , eND corresponde à borda não-direcional.

A orientação de borda dominante é determinada a partir destas forças de borda, e corresponde àquela que possui o indicador de orientação com o valor mais elevado, conforme apresentado na Eq. (4.7):

$$DomOr = \arg \max_{Or \in \{V, H, 45^\circ, 135^\circ, ND\}} \{eOr\} \quad (4.7)$$

Para PUs com tamanho maior do que 4x4, tal como a PU 16x16 da Figura 4.6, um procedimento ligeiramente diferente é utilizado. Inicialmente, os cálculos de borda de

todas as PUs 4x4 que compõem a PU maior são efetuados. Logo após, a direção com o maior valor médio é definida como a orientação de borda dominante da PU maior.

O segundo passo do algoritmo proposto é a definição de quais modos devem ser avaliados no processo de predição intra da PU atual, considerando a orientação de borda dominante. Inicialmente, quatro conjuntos de modos angulares foram definidos a partir do conjunto completo de 33 modos angulares utilizados no HEVC. Para compor esses conjuntos, os 33 modos foram categorizados em quatro grupos conforme a orientação da borda: horizontal, vertical, diagonal 45° e diagonal 135°.

Cada conjunto associado a cada orientação de borda, inicialmente era composto por quinze modos angulares, conforme apresentado na Figura 4.7. A composição destes conjuntos é apresentada nas equações de (4.8) a (4.11):

$$eVerSet = \{19,20,21,22,23,24,25,26,27,28,29,30,31,32,33\} \quad (4.8)$$

$$eHorSet = \{3,4,5,6,7,8,9,10,11,12,13,14,15,16,17\} \quad (4.9)$$

$$e45^{\circ}Set = \{27,28,29,30,31,32,33,34,2,3,4,5,6,7,8\} \quad (4.10)$$

$$e135^{\circ}Set = \{1,12,13,14,15,16,17,18,19,20,21,22,23,24,25\} \quad (4.11)$$

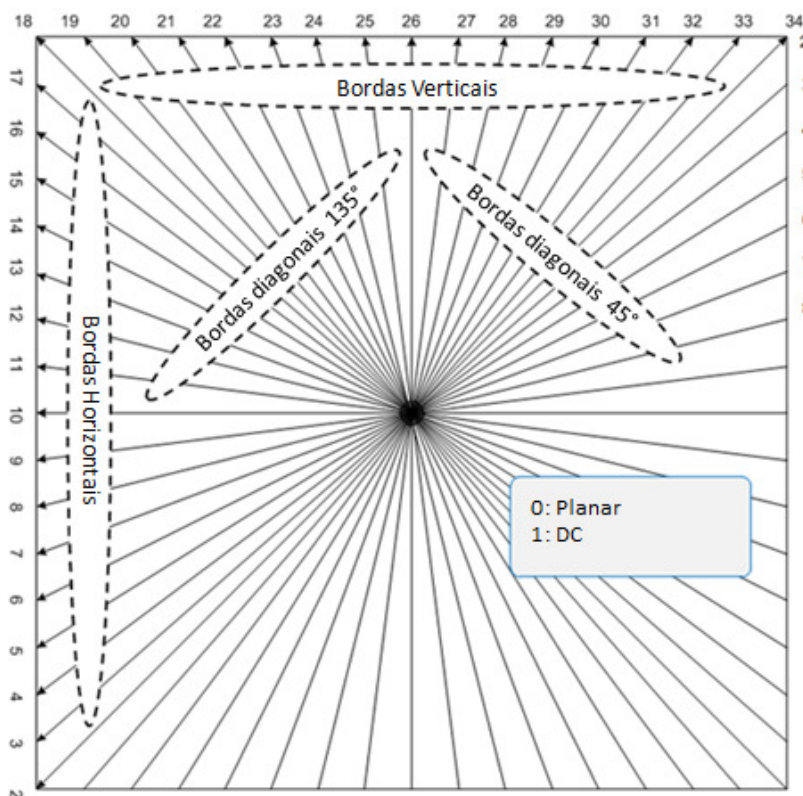


Figura 4.7: Conjuntos de orientação de borda.

A partir dessa categorização em quatro conjuntos de modos de acordo com a orientação de borda, um grupo de experimentos foi realizado utilizando as condições comuns de teste recomendadas em [43] usando a configuração *all-intra* (AI) e quatro parâmetros de quantização (QPs): 22, 27, 32 e 37, conforme explicado a seguir.

Num primeiro experimento, para cada orientação de borda, os oito modos mais frequentemente selecionados para cada uma das quatro orientações descritas acima (vertical, horizontal, diagonal 45° e diagonal 135°) foram incluídos num subconjunto associado a essa orientação. A seguir, experimentos semelhantes foram realizadas com sete, nove e dez modos angulares por subconjunto.

A análise dos resultados de desempenho e complexidade de codificação demonstrou que o subconjunto composto por nove modos alcançou o melhor compromisso entre a eficiência de codificação e a redução da complexidade, conforme apresentado nas Figuras 4.8 e 4.9.

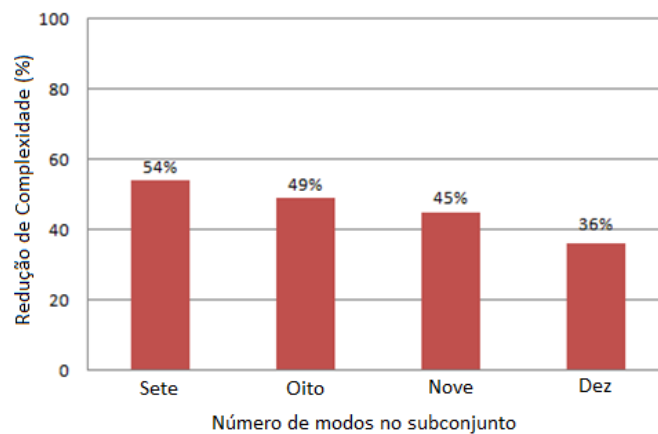


Figura 4.8: Redução de Complexidade para cada tamanho de subconjunto avaliado.

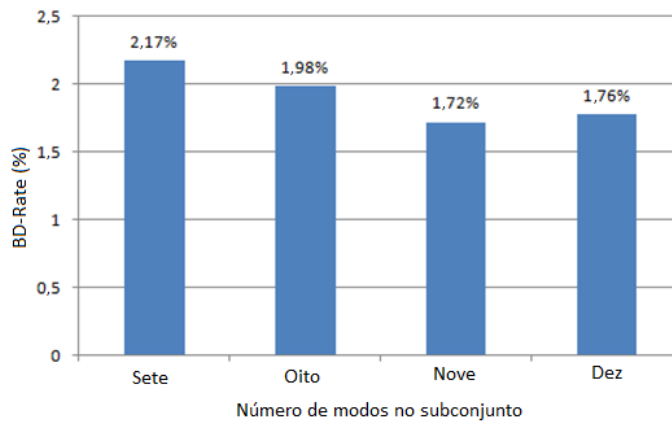


Figura 4.9: BD-Rate para cada tamanho de subconjunto avaliado.

Os gráficos apresentados nas Figuras 4.8 e 4.9 apresentam os valores médios para todos os experimentos avaliados. A partir destas figuras, concluiu-se que uma redução de complexidade considerável (45%) no processo de predição intra foi atingida quando o subconjunto de nove modos foi utilizado, apresentando uma pequena degradação no desempenho (1,72%), conforme medido pelo *BD-rate* [40][41]. Por esta razão, este tamanho de subconjunto foi escolhido para ser utilizado no algoritmo proposto.

Cada subconjunto de nove modos angulares foi associado com uma orientação de borda (horizontal, vertical, 45°, 135°). O subconjunto não-direcional foi definido para ser usado quando a textura da PU não tem uma orientação dominante. Este subconjunto foi composto a partir dos modos mais frequentemente escolhidos de cada subconjunto direcional.

Para compor os quatro subconjuntos direcionais de nove modos, um grupo de experimentos foi realizado utilizando as condições de teste recomendadas em [43] com aplicação da configuração *all-intra*. Nestes experimentos, os quatro conjuntos de quinze modos (definidos anteriormente) foram avaliados e para cada orientação de borda, os nove modos mais frequentemente escolhidos foram incluídos no subconjunto associado com aquela direção.

Os histogramas das Figuras 4.10 a 4.14 apresentam, no eixo horizontal, os índices dos quinze modos intra associados a cada orientação de borda e, no eixo vertical, as suas frequências médias de utilização (com valores normalizados). Os índices destacados nas Figuras 4.10 a 4.14 indicam os modos mais frequentes que serão associados com aquela orientação de borda específica.

Na Figura 4.10, os modos de 22 a 30, em destaque, irão compor o subconjunto associado com as bordas orientadas verticalmente.

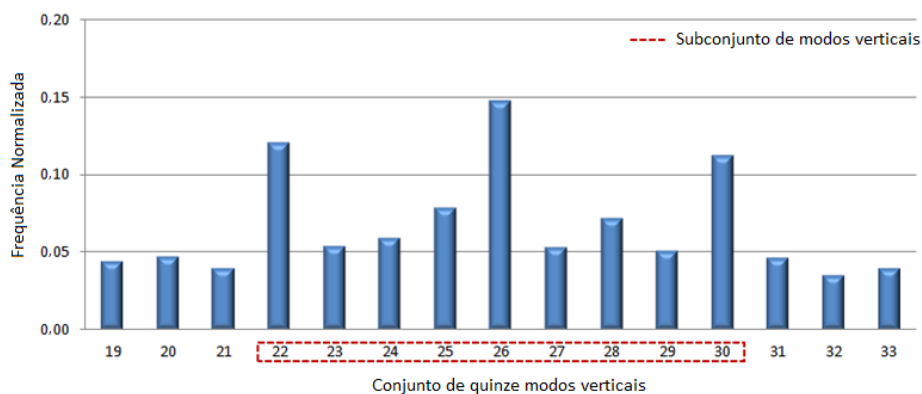


Figura 4.10: Conjunto de modos vertical versus frequência de utilização normalizada.

Na Figura 4.11, os modos de predição de 6 a 14 compõem o subconjunto de modos associados com as bordas horizontais. A Figura 4.12 e a Figura 4.13 destacam os nove modos que compõem os subconjuntos diagonais 45° e 135°, respectivamente.

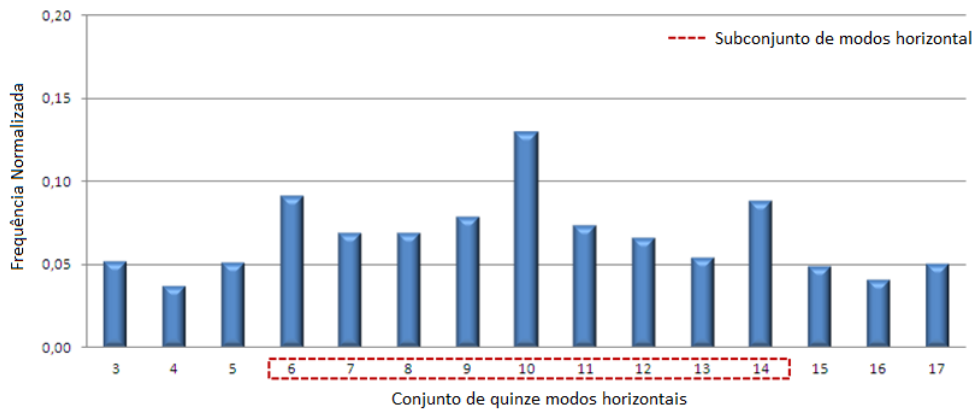


Figura 4.11: Conjunto de modos horizontal versus frequência de utilização normalizada.

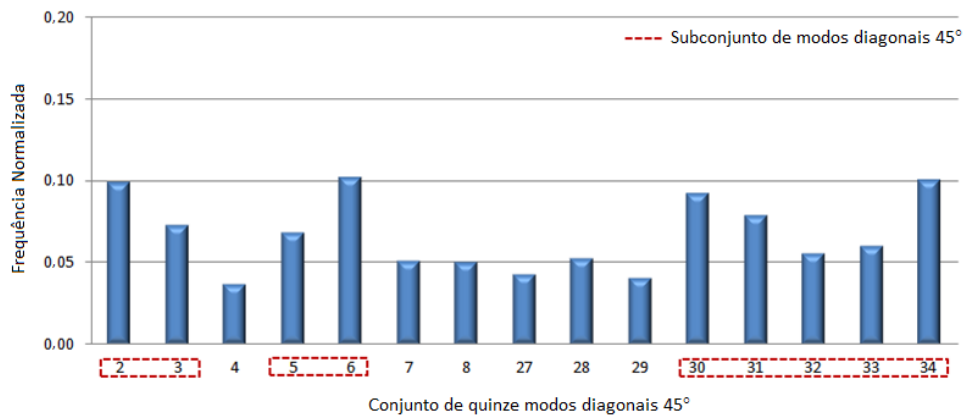


Figura 4.12: Conjunto de modos diagonal 45° versus frequência de utilização normalizada.

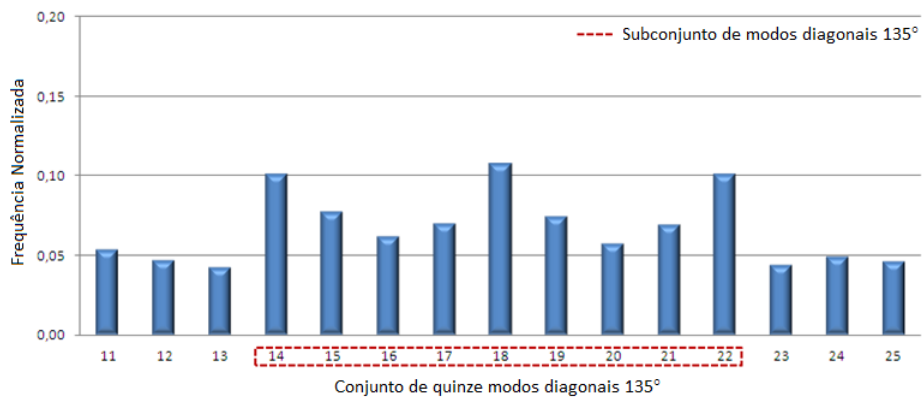


Figura 4.13: Conjunto de modos diagonal 135° versus frequência de utilização normalizada.

Na Figura 4.14, o eixo horizontal do histograma apresenta todos os modos intra adotados pelo HEVC, destacando os modos escolhidos para o subconjunto não-direcional, enquanto o eixo vertical mostra as frequências de utilização.

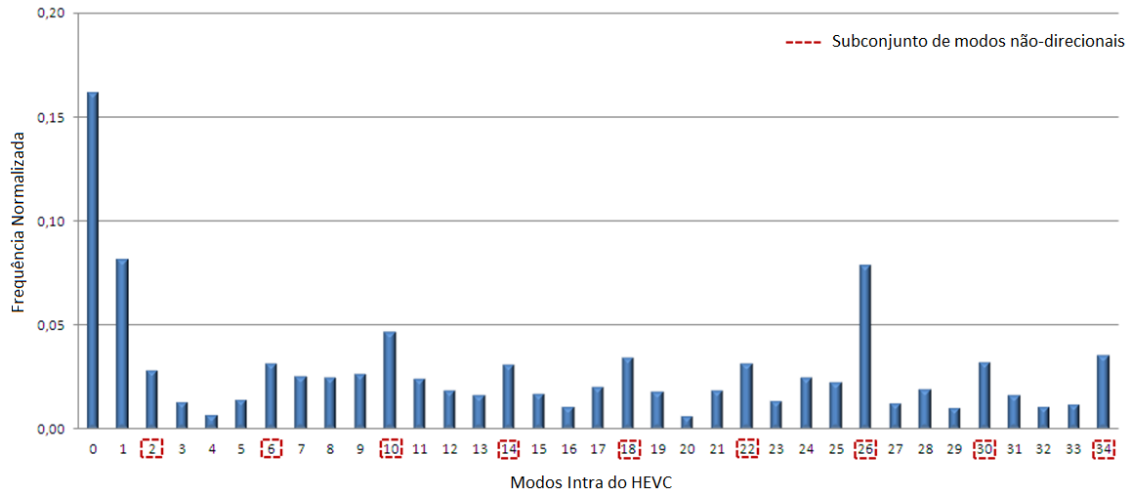


Figura 4.14: Conjunto de modos intra versus frequência de utilização normalizada.

Assim, a partir desta análise, cinco subconjuntos com nove modos angulares foram formados, cuja composição é apresentada nas equações (4.12) a (4.16), e também é graficamente representados na Figura 4.15.

$$eVerSubset = \{22,23,24,25,26,27,28,29,30\} \quad (4.12)$$

$$eHorSubset = \{6,7,8,9,10,11,12,13,14\} \quad (4.13)$$

$$e45^\circ Subset = \{30,31,32,33,34,2,3,5,6\} \quad (4.14)$$

$$e135^\circ Subset = \{14,15,16,17,18,19,20,21,22\} \quad (4.15)$$

$$eNDSUBset = \{2,6,10,14,18,22,26,30,34\} \quad (4.16)$$

O subconjunto não-direcional foi composto a partir dos três modos mais frequentemente escolhidos de cada subconjunto direcional (modos 22, 26 e 30 a partir subconjunto vertical; modos 6, 10 e 14 a partir do subconjunto horizontal; modos 30, 34 e 2 a partir do subconjunto 45° e; modos 14, 18 e 22 a partir do subconjunto 135°).

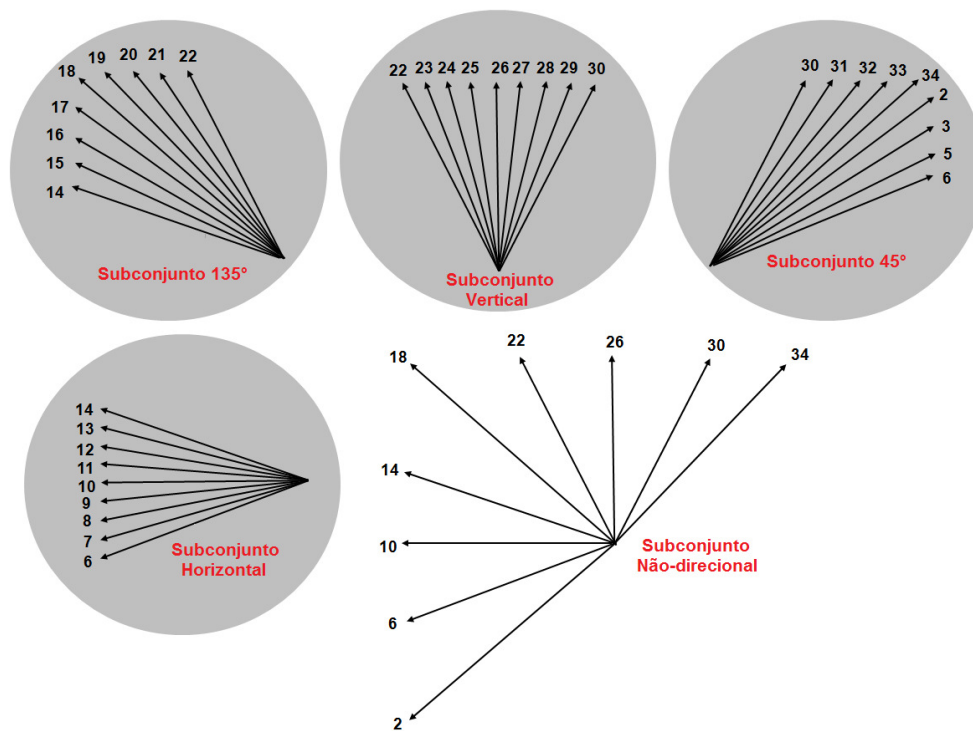


Figura 4.15: Subconjuntos de modos para cada orientação de borda.

Durante o processo de predição intra somente um desses cinco subconjuntos é utilizado para codificar cada PU, o qual é selecionado de acordo com a orientação de borda dominante, *DomOr*, definida pela equação (4.7). Apenas os modos de predição pertencentes a esse subconjunto serão avaliados durante a pesquisa RDO para encontrar o melhor modo de predição. Devido a sua importância e elevada frequência de ocorrência, os modos DC e Planar sempre são avaliados, independentemente de qual dos cinco subconjuntos for selecionado.

A Figura 4.16 apresenta uma trama, exemplificando tipos de borda dominante de acordo com a textura da imagem. Na figura são apresentados os tipos: Horizontal, Vertical e 135°.

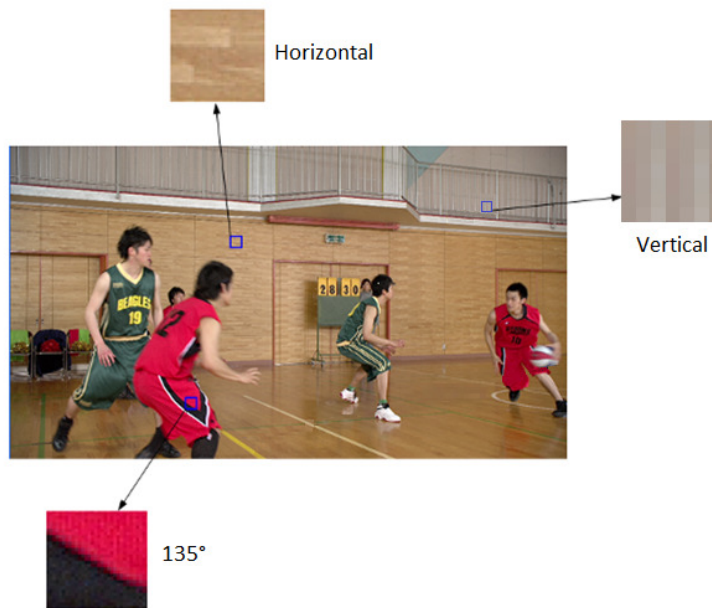


Figura 4.16 : Exemplo de tipos de bordas dominantes de acordo com a textura da PU.

A Figura 4.17 apresenta um fluxograma detalhado do algoritmo para a decisão de modo apresentado nesta seção.

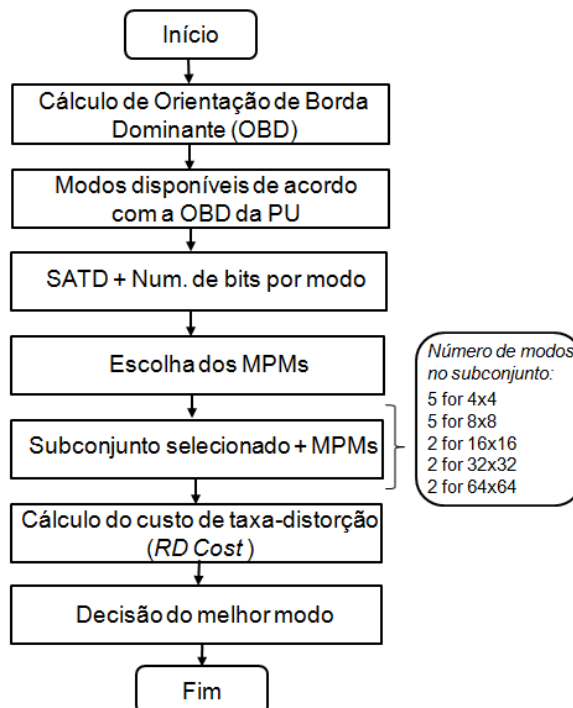


Figura 4.17: Fluxograma do algoritmo proposto para a decisão de modo intra baseado na orientação de borda dominante.

Para avaliar a redução de complexidade computacional realizada quando o algoritmo proposto é utilizado e o seu desempenho de codificação em comparação com o HEVC original, um grupo de experimentos foi realizado utilizando sequências de teste

especificadas nas condições comuns de teste da JCT-3V [86], com três vista cada uma. Tais sequências foram utilizadas pois, posteriormente, os algoritmos apresentados nessa seção serão utilizados em um cenário multivista, o que facilitará as comparações de desempenho e complexidade computacional atribuída. Detalhes destas sequências são apresentados na Tabela 4.1 e mais informações são apresentados no Apêndice B desta tese.

A predição intra incluindo o algoritmo de orientação borda dominante, proposto nesta seção, foi implementada usando o software de referência HM10 do HEVC [42] para fazer essa avaliação.

Tabela 4.1: Características das sequências de vídeo.

Sequência	Taxa de Tramas (<i>fps</i>)	Número de Tramas	Resolução (pixels)
Balloons	30	100	1024x768
Newspaper	30	100	1024x768
Kendo	30	100	1024x768
GTFly	25	100	1920x1080
PoznanStreet	25	100	1920x1080
UndoDancer	25	100	1920x1080

Para comparar o desempenho e a complexidade do algoritmo proposto em relação ao HM10 alguns indicadores de desempenho foram avaliados nomeadamente: percentagem média da diferença na taxa de bits ($\Delta Taxa\ de\ Bits$), percentagem média do tempo de codificação ($\Delta Tempo$) e a diferença média do PSNR de luminância ($\Delta PSNR$), os quais foram definidos pelas equações (4.17) a (4.19).

$$\Delta PSNR_Y = PSNR_Y_{proposto} - PSNR_Y_{HM10.0} \quad (4.17)$$

$$\Delta Taxa de Bits = \frac{(Taxa de Bits_{proposto} - Taxa de Bits_{HM10.0})}{Taxa de Bits_{HM10.0}} \times 100 \quad (4.18)$$

$$\Delta Tempo = \frac{Tempo_{proposto} - Tempo_{HM10.0}}{Tempo_{HM10.0}} \times 100 \quad (4.19)$$

Além disso, as medidas de *BD-PSNR/Rate* [40][41] apresentadas na Seção 2.3, também foram utilizadas para avaliar os resultados em termos eficiência de codificação. A

plataforma de simulação utilizada consiste em servidores baseados em CPUs Intel® Xeon® E5520 CPU 2.27GHz, com 24 GB de RAM e correndo Windows Server 2008. O software HM foi compilado usando Visual C++.

De acordo com os resultados das simulações apresentados na Tabela 4.2, o esquema proposto neste algoritmo permite atingir uma redução no tempo de codificação de 34,88% em relação ao HM10, com uma perda insignificante em termos de eficiência de codificação (0,08 dB em BD-PSNR, em média).

Tabela 4.2: Comparação do algoritmo de Orientação de borda dominante proposto em relação ao algoritmo usado no HM10.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNR Y [dB]
<i>Balloons</i>	25	0,77	-0,03	-34,51	1,6	-0,10
	30	0,79	-0,04	-36,73		
	35	1,04	-0,05	-37,51		
	40	1,25	-0,04	-40,54		
<i>Newspaper</i>	25	0,81	-0,01	-31,68	1,5	-0,07
	30	0,89	-0,03	-35,71		
	35	1,11	-0,03	-35,47		
	40	1,35	-0,03	-37,17		
<i>Kendo</i>	25	0,83	-0,03	-32,16	1,7	-0,09
	30	0,88	-0,03	-35,17		
	35	1,19	-0,03	-36,30		
	40	1,46	-0,04	-33,79		
<i>GTFly</i>	25	0,79	-0,02	-30,93	1,7	-0,06
	30	1,02	-0,02	-34,69		
	35	1,13	-0,03	-35,21		
	40	1,39	-0,03	-35,61		
<i>PoznanStreet</i>	25	0,64	-0,02	-30,27	1,7	-0,07
	30	0,89	-0,02	-33,98		
	35	1,24	-0,03	-36,88		
	40	1,39	-0,04	-37,91		
<i>UndoDancer</i>	25	0,88	-0,02	-30,69	1,6	-0,06
	30	0,90	-0,02	-32,53		
	35	1,13	-0,02	-34,82		
	40	1,29	-0,02	-36,91		
Média		1,04	-0,028	-34,88	1,63	-0,075

A Tabela 4.2 apresenta os resultados referentes a primeira vista de cada sequência, uma vez que as três vistas de cada sequência de teste apresentam resultados muito semelhantes. Considerando os resultados obtidos, concluiu-se que o algoritmo proposto reduz a complexidade da predição intra do HEVC em cerca de 35% em média, com uma perda mínima em desempenho de 0,075 dB em média em BD-PSNR, o que nos leva a considerar este algoritmo uma base promissora para as demais otimizações propostas nas próximas seções.

4.1.2. Correlação entre as PUs de Níveis de Profundidade Adjacentes

Dando continuidade às explorações anteriores, uma característica adicional do codificador HEVC foi analisada com o objetivo de reduzir ainda mais a complexidade computacional do processo de predição intra. Essa característica adicional consiste em explorar as correlações existentes entre os níveis de profundidade da árvore de codificação hierárquica adotada pelo HEVC. Tais explorações permitiram verificar que a correlação entre a informação de borda da PU atual, a qual está sendo codificada, e a informação de borda das PUs já codificadas em níveis anteriores de profundidade da árvore é elevada.

Esta informação de borda é útil pois, de acordo com a orientação de borda (horizontal, vertical, 45°, 135° ou não-direcional) da textura da PU atual, e as orientações de borda das PUs de níveis adjacentes, é possível verificar se estas PUs têm a mesma orientação dominante e, em caso afirmativo, os modos candidatos utilizados nas PUs já codificadas podem ser reusados para codificar a PU atual, conforme descrito em mais detalhe a seguir.

A Figura 4.18 apresenta a estrutura de árvore hierárquica das PUs, ilustrando o reuso do subconjunto de modos candidatos utilizado na PU do nível de profundidade anterior para codificar a PU atual. Tal reuso ocorre quando a orientação de borda dominante da PU atual é a mesma orientação de borda da PU predita anteriormente.

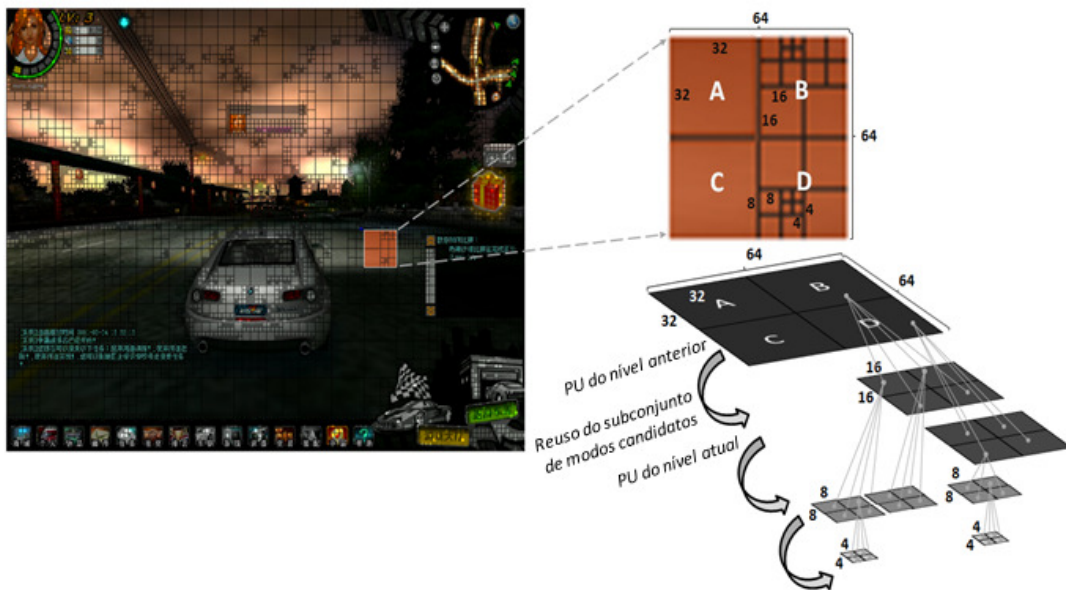


Figura 4.18: Estrutura de árvore hierárquica das PUs no HEVC.

A exploração apresentada nesta seção começou por um estudo, no qual as sequências de teste: ‘PoznanHall2’, ‘Shark’, ‘PoznanCarPark’ e ‘PoznanHall1’ especificadas em [86] e [102], foram codificadas com os parâmetros de quantização 25, 30, 35 e 40. Neste estudo, as frequências de uso de cada modo de predição intra foram analisadas em busca de correlações entre os modos intra escolhidos em PUs de níveis adjacentes da árvore. É importante ressaltar que a correlação entre os níveis de profundidade da árvore para todos os tamanhos de PU (PUs 64x64, 32x32, 16x16, 8x8 e 4x4) foram avaliados.

Esta análise procurou determinar se o melhor modo escolhido para a PU atual possuía a mesma orientação de borda (por exemplo: orientação vertical), que o modo intra escolhido na PU do nível profundidade imediatamente inferior, ou seja, se o modo selecionado na PU atual pertencia a um subconjunto de modos com a mesma orientação que o modo intra da PU codificada no nível anterior de profundidade da árvore. Os resultados desta análise, considerando-se a média dos resultados para todas as sequências de vídeo testadas, são apresentados na Tabela 4.3.

Tabela 4.3: Correlação na orientação de borda das PUs inter-níveis.

QP	Mesma Orientação (%)	Orientação Diferente (%)
25	91,90	8,10
30	90,06	9,94
35	87,78	12,22
40	86,12	13,88
Med.	88,97	11,03

A partir destes resultados, foi possível observar que, na maioria dos casos, o modo intra escolhido como o melhor modo da PU atual, fazia parte do mesmo subconjunto de modos angulares (com a mesma orientação) que o modo intra selecionado na PU de nível de profundidade imediatamente inferior.

Esta observação motivou a introdução de um novo teste para verificar se a PU atual tem a mesma orientação de borda dominante que a PU de profundidade anterior, tal como ilustrado no terceiro passo do fluxograma do algoritmo, na Figura 4.19.

Se o resultado do teste for positivo, tendo em conta as correlações acima referidas, considera-se que o subconjunto de modos candidatos usado para codificar a PU da profundidade anterior é uma boa escolha para codificar a PU atual e, então esse subconjunto é propagado (ou reusado) no próximo nível de codificação da árvore, tal como ilustrado na Figura 4.18. Além disso, quando estas orientações de borda são iguais, os passos seguintes do processo de predição, que correspondem ao cálculo de SATD e escolha dos MPMs, podem ser evitados e o cálculo de custo RD é imediatamente realizado, conforme apresentado na Figura 4.19.

No entanto, se o reuso do subconjunto de modos candidatos não ocorre, o quarto e quinto passos do algoritmo são realizados. Nesses passos, o cálculo de SATD é efetuado e logo após, no passo seguinte a escolha de MPMs é feita. Estes dois passos estão descritos na Seção 4.1. No sexto passo, o subconjunto de modos candidatos é definido de acordo com o tamanho da PU e os MPMs escolhidos são adicionados a este subconjunto.

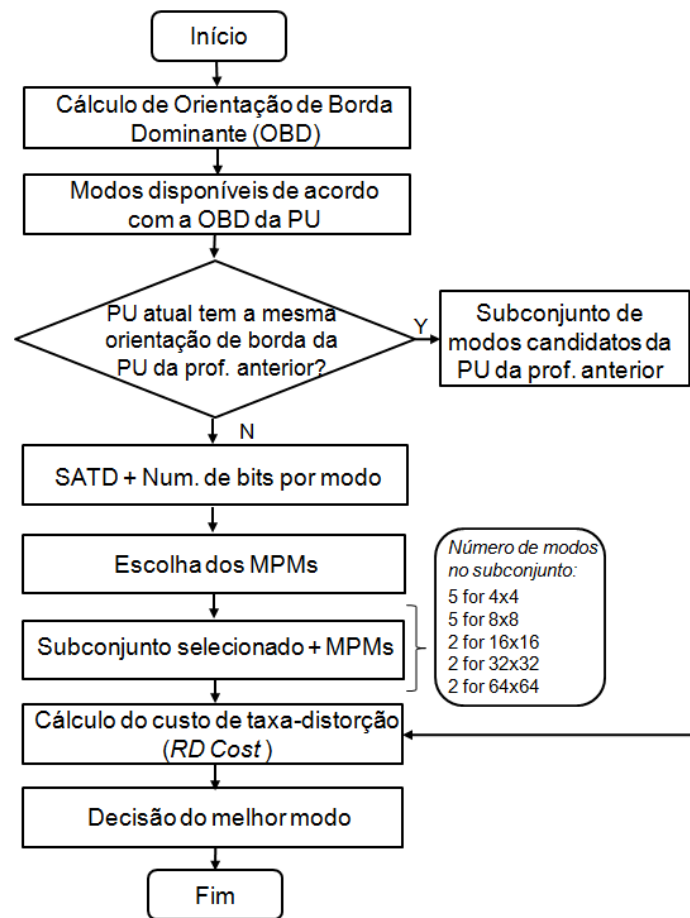


Figura 4.19: Fluxograma do algoritmo proposto para a decisão de modo intra baseado na orientação de borda e na estrutura de árvore hierárquica do HEVC.

A Tabela 4.4 apresenta os resultados do algoritmo proposto, o qual explora a correlação entre a orientação de borda de PUs de níveis adjacentes de profundidade da árvore de codificação. É importante ressaltar que o algoritmo proposto nessa seção inclui a otimização proposta pelo algoritmo apresentado na seção anterior, e tem como objetivo simplificar ainda mais a decisão de modo da predição intra do HEVC.

Tabela 4.4: Comparação do algoritmo de Correlação entre a Orientação de Borda das PUs Inter-níveis proposto em relação ao algoritmo usado no HM10.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	0,72	-0,04	-35,92	1,6	-0,11
	30	0,80	-0,04	-36,10		
	35	1,03	-0,05	-35,36		
	40	1,22	-0,05	-37,33		
<i>Newspaper</i>	25	0,85	-0,01	-32,44	1,5	-0,07
	30	0,92	-0,02	-35,56		
	35	1,12	-0,03	-37,02		
	40	1,60	-0,03	-40,17		
<i>Kendo</i>	25	0,87	-0,03	-34,64	1,8	-0,09
	30	0,95	-0,03	-37,17		
	35	1,20	-0,03	-38,49		
	40	1,39	-0,03	-36,21		
<i>GTFly</i>	25	0,74	-0,01	-35,95	1,8	-0,06
	30	1,10	-0,02	-36,21		
	35	1,31	-0,02	-38,80		
	40	1,59	-0,02	-37,25		
<i>PoznanStreet</i>	25	0,53	-0,01	-34,23	1,6	-0,06
	30	0,77	-0,02	-36,39		
	35	1,37	-0,02	-39,98		
	40	1,43	-0,05	-39,65		
<i>UndoDancer</i>	25	0,94	-0,02	-32,57	1,8	-0,07
	30	0,97	-0,03	-35,02		
	35	1,10	-0,02	-36,87		
	40	1,45	-0,02	-39,39		
Média		1,08	-0,027	-36,61	1,68	-0,08

A partir dos resultados da Tabela 4.4, é possível verificar que o tempo de processamento da codificação em relação ao HM10 foi reduzido em 36,61%, em média, com uma perda mínima de 0,08dB em BD-PSNR e cerca de 1,7% em *BD-rate*. Isto significa que a inserção desta funcionalidade ao algoritmo rápido para a decisão do modo intra, reduziu o tempo de processamento em mais 1,73% em média, com uma perda insignificante em eficiência de codificação de 0.005dB considerando o BD-PSNR.

4.1.3. Verificação de Modo-Fronteira

Buscando aperfeiçoar ainda mais o processo de predição intra-tramas do HEVC, propôs-se e avaliou-se uma modificação adicional ao método de decisão do modo intra, conforme descrito a seguir.

O algoritmo proposto nessa seção considera a eventual adição de mais um modo ao subconjunto de modos candidatos selecionado, de forma que, dependendo dos modos selecionados depois do cálculo de SATD, uma verificação adicional é realizada pelo algoritmo conforme apresentado a seguir.

Nesta verificação, um teste é efetuado para averiguar se algum dos modos candidatos selecionados corresponde a um modo-fronteira, o qual é um modo que faz fronteira entre dois subconjuntos de modos vizinhos.

Se essa condição se verificar, o modo de predição direcional mais próximo e pertencente ao subconjunto de direções vizinho àquele que possui o modo-fronteira, é adicionado ao subconjunto modos candidatos. Este processo é ilustrado na Figura 4.20, onde os modos de 14, 22 e 30 são modos-fronteira, e as setas tracejadas indicam a adição de um dos seguintes modos: 13, 15, 21, 23, 29 ou 31 ao subconjunto de modos candidatos.

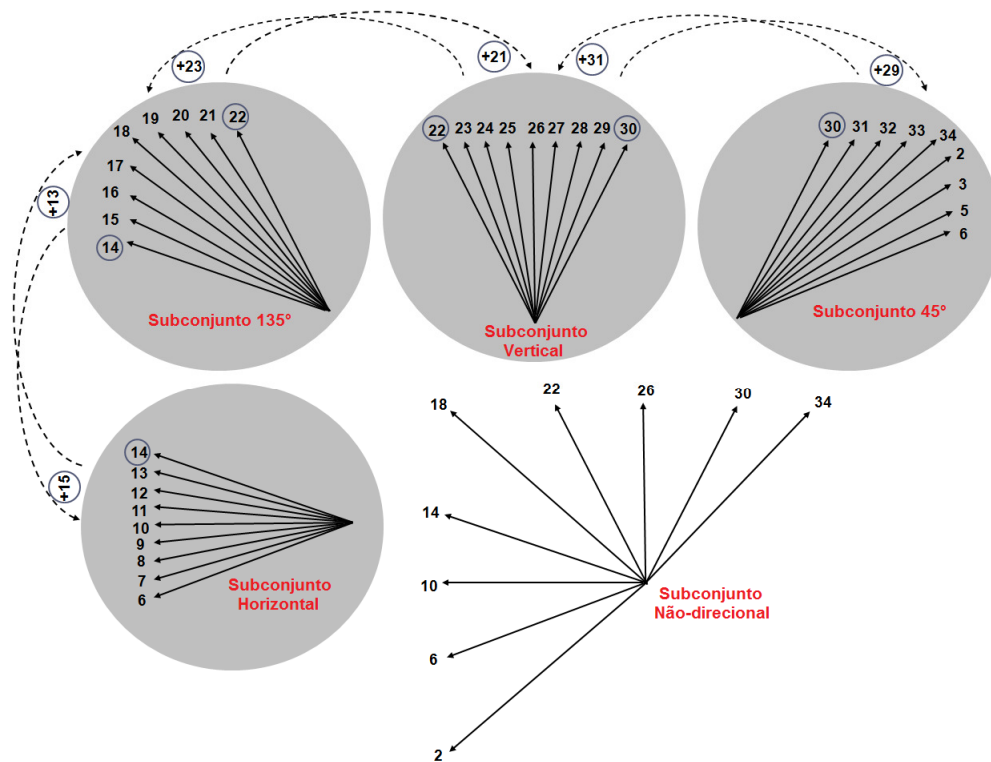


Figura 4.20: Subconjunto de modos para cada orientação de borda incluindo a representação do teste de modo-fronteira.

Um exemplo desse processo de verificação do modo-fronteira e adição de um modo ao subconjunto de candidatos, pode ser constatado quando a orientação de borda 135° é calculada como a orientação de borda dominante da PU e o modo 22 (modo-fronteira) é escolhido como um modo candidato (no passo "Subconjunto selecionado +

MPM" da Figura 4.20), então o modo 23 (do subconjunto vertical) é adicionado ao subconjunto de modos candidatos a serem testadas no processo RDO.

A Figura 4.21 apresenta o fluxograma referente ao algoritmo descrito nesta seção, a partir do qual é possível observar que, logo após a seleção do subconjunto de modos candidatos, um teste é realizado para verificar se algum modo-fronteira faz parte desse subconjunto e, em caso positivo, um modo suplementar é adicionado ao subconjunto de modos candidatos.

Após o teste de modo-fronteira, os últimos passos do algoritmo são o cálculo do custo RD e a decisão do melhor modo de predição intra, conforme descrito na Seção 4.1.

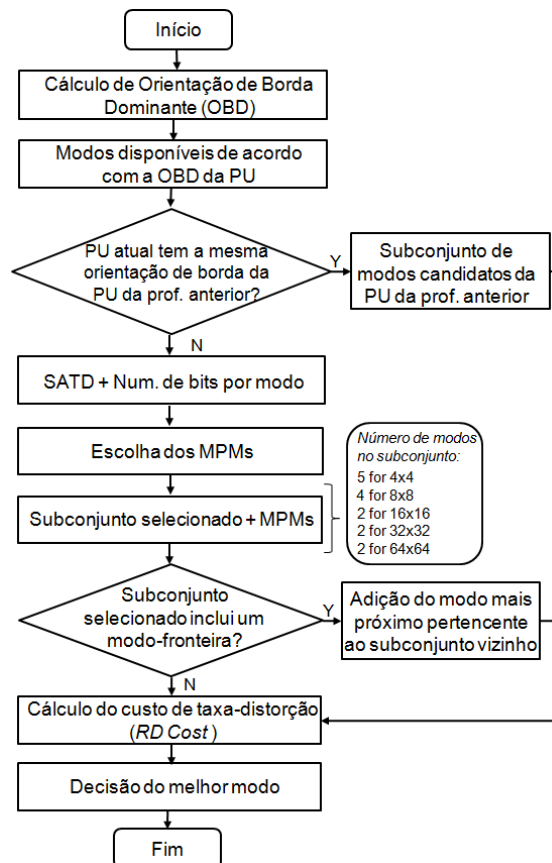


Figura 4.21: Fluxograma do algoritmo proposto para a decisão de modo intra utilizando a verificação de modo-fronteira.

A Tabela 4.5 apresenta os resultados do algoritmo proposto, o qual inclui as otimizações apresentadas nos algoritmos das seções anteriores e adiciona ainda uma verificação de modo-fronteira, visando incrementar a redução de complexidade computacional alcançada pelos demais algoritmos.

Tabela 4.5: Comparação do algoritmo de Verificação do modo-fronteira proposto em relação ao algoritmo usado no HM10.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	0,85	-0,04	-37,91	1,7	-0,11
	30	0,91	-0,04	-39,24		
	35	1,14	-0,04	-38,74		
	40	1,34	-0,05	-39,31		
<i>Newspaper</i>	25	0,92	-0,02	-36,94	1,6	-0,08
	30	1,04	-0,03	-38,85		
	35	1,24	-0,03	-40,35		
	40	1,40	-0,03	-42,83		
<i>Kendo</i>	25	0,76	-0,03	-35,79	1,8	-0,09
	30	0,92	-0,03	-39,69		
	35	1,14	-0,03	-39,03		
	40	1,48	-0,04	-38,81		
<i>GTFly</i>	25	0,67	-0,02	-36,95	1,7	-0,06
	30	0,89	-0,03	-38,61		
	35	1,05	-0,03	-39,72		
	40	1,16	-0,03	-40,44		
<i>PoznanStreet</i>	25	0,76	-0,02	-38,58	1,8	-0,07
	30	0,91	-0,03	-39,84		
	35	1,19	-0,03	-41,08		
	40	1,33	-0,04	-42,23		
<i>UndoDancer</i>	25	0,87	-0,02	-36,41	1,7	-0,07
	30	0,98	-0,02	-36,74		
	35	1,10	-0,02	-39,51		
	40	1,31	-0,02	-41,35		
Média		1,06	-0,030	-39,12	1,72	-0,08

A partir desses resultados, é possível verificar uma redução de complexidade de 39,12%, em média, com uma ligeira degradação em BD-PSNR de 0,08dB e um aumento de 1,72% em *BD-rate*. Isto significa que, com a inserção do novo teste, foi possível reduzir o tempo de processamento em mais 2,51% em média sem degradação adicional no desempenho de codificação, quando comparado com a solução apresentada na seção anterior.

4.2. Análise de Resultados e Comparações

Esta seção apresenta uma análise dos resultados obtidos neste capítulo, seguida de comparações com trabalhos relacionados. A solução apresentada na Seção 4.1.1, descreve um método rápido e eficiente para a predição intra do HEVC, o qual calcula a orientação de borda dominante da textura das PUs para definir os modos mais relevantes a serem avaliados no processo de decisão de modo da predição intra do HEVC. Logo após, uma otimização para o modo de decisão intra é apresentada na Seção 4.1.2, a qual explora a correlação entre os modos intra de PUs de níveis de profundidade adjacentes da árvore. E finalmente, na Seção 4.1.3, um método de predição intra rápido incluindo um refinamento adicional foi implementado. Neste refinamento um teste de modo-fronteira é realizado durante o processo de composição do subconjunto de modos intra candidatos que serão avaliados no cálculo de custo de taxa de distorção. E, caso o subconjunto de modos candidatos possua um modo-fronteira, um modo de predição intra angular é adicionado a este subconjunto.

Os resultados experimentais demonstraram que, quando comparado com o software de referência do HEVC, a solução inicial que explorou a orientação de borda dominante apresentou uma redução média em complexidade de 35%, com uma perda mínima em desempenho de 0,075dB em média em BD-PSNR, sendo utilizada como base para as implementações propostas nas seções seguintes. A seguir, a partir dos resultados obtidos acrescentando-se a análise e exploração das correlações entre os modos intra das PUs de níveis de profundidade vizinhos na árvore de codificação, foi possível verificar uma redução no tempo de processamento da codificação intra de 36,61%, em média, com uma perda mínima de 0,08dB em BD-PSNR. Finalmente, após a otimização apresentada na Seção 4.1.3, a qual adotou a verificação do modo-fronteira e adição condicional de um modo suplementar ao subconjunto de candidatos, o método rápido para a decisão do modo intra proposto alcançou uma redução em complexidade computacional de aproximadamente 40% em média, com uma perda mínima em BD-PSNR de 0,08 dB em média.

Comparando-se o método de codificação intra rápido apresentado na Seção 4.1.3, com os trabalhos relacionados descritos na Seção 4.1, é possível concluir que utilizando o método proposto nessa tese, uma maior redução no tempo de processamento do codificador é alcançada, em comparação com a redução apresentada por Zhao et al.

[78]. Tendo em vista que o método desenvolvido por Zhao et al. [78] foi parcialmente adotado pelo HM desde a versão 2.0, e o algoritmo proposto no presente trabalho foi testado utilizando o HM versão 10.0 (o qual já inclui o procedimento inserido no HM 2.0) o algoritmo aqui proposto aumentou ainda mais a economia de tempo de codificação introduzida pelo método de Zhao et al. [78].

Em termos de redução do tempo de codificação, o algoritmo proposto neste trabalho ultrapassa a redução média de 22,99% apresentada por Kim et al. [79]. O algoritmo proposto supera também a redução de complexidade computacional relatada por Jiang et al. [80], o qual reduz o tempo de computação da predição intra em 19,99%, em média. No que diz respeito à eficiência de codificação, o algoritmo apresentado por Jiang et al. [80] apresenta um aumento médio de 0,74% na taxa de bits e uma diminuição de 0.04dB em PSNR quando comparado com o HM4.0. Quando comparado com o algoritmo proposto por Tian e Goto [81], o algoritmo proposto no presente trabalho supera a redução de complexidade de 30,71% apresentada por eles quando a configuração *all-intra* de alta eficiência é utilizada. O método proposto neste capítulo apresenta melhores resultados do que os apresentados em Kim et al. [82], o qual economiza em média 24% no tempo de codificação quando a configuração *all-intra* é aplicada.

Quando comparado com Zhang et al. [83], o trabalho proposto excede a redução na complexidade alcançada por aqueles autores, a qual corresponde a cerca de 15% para a configuração *all-intra*. Em termos de desempenho, Zhang et al. [83] apresenta um aumento no *BD-rate* de 0,64%, em média, quando comparado com o HM4.0.

Quando comparado com Zhang e Ma [84], a solução proposta aqui apresenta resultados semelhantes em termos de redução da complexidade (38% em média). No entanto, em relação a eficiência de codificação, o nosso trabalho supera Zhang e Ma [84], o qual relata um aumento no *BD-rate* de 2,9% em média, enquanto este trabalho apresentou um aumento de 1,72% em média, considerando o *BD-rate*.

A comparação dos resultados de redução de complexidade computacional obtidos neste trabalho em comparação com aqueles apresentados nos trabalhos relacionados estão sumarizadas na Tabela 4.6

Tabela 4.6: Comparação da redução de complexidade do algoritmo proposto em relação aos trabalhos relacionados

Trabalho	Redução de Complexidade [%]
Zhao et al., 2011 [78]	20
Kim et al., 2013 [79]	24
Jiang et al., 2012 [80]	19,99
Tian and Goto, 2012 [81]	30,71
Kim et al., 2011 [82]	22,99
Zhang et al., 2012 [83]	15
Zhang and Ma, 2012 [84]	38
Algoritmo Proposto	39,12

A partir desta tabela, é possível observar que o método para a decisão do modo de predição intra do HEVC proposto neste trabalho apresenta a maior redução de complexidade computacional comparado aos trabalhos relacionados, quando a configuração do *all-intra* é aplicada.

O próximo capítulo (Capítulo 5) apresenta algoritmos rápidos para a predição inter-vista do modo intra, e predição inter-vista da profundidade da árvore de codificação. Além disso, métodos de aceleração da codificação intra do 3D-HEVC são apresentados, visando reduzir a complexidade de codificação dos mapas de profundidade e vistas de textura.

5. ALGORITMOS RÁPIDOS PARA CODIFICAÇÃO INTRA DA EXTENSÃO 3D-HEVC

Na codificação de vídeo multivista, além das previsões inter e intra-trama, as similaridades entre vistas de câmeras adjacentes são também exploradas visando melhorar a eficiência de compressão quando comparada à codificação independente das vistas [87][88].

No entanto, conforme os resultados experimentais apresentados em [89], quando comparamos a codificação independente das vistas utilizando o HEVC *Simulcast*, com a codificação multivista utilizando o MVC (extensão multivista do H.264/AVC), o HEVC *Simulcast* supera a economia de *bitrate* do MVC, atingindo uma eficiência de codificação superior [89], o que motivou o desenvolvimento dos algoritmos apresentados na Seção 5.1 a seguir.

5.1. HEVC *Simulcast* vs. HEVC Multivista

A arquitetura do HEVC Multivista ou MV-HEVC (extensão 3D-HEVC sem mapas de profundidade) [11] é conceitualmente similar e utiliza os mesmos princípios da extensão MVC do H.264/AVC [13]. As ferramentas básicas de codificação utilizadas no HEVC Multivista são idênticas às do HEVC e o processo de previsão intra/inter é o mesmo utilizado pelo padrão HEVC [13].

Um dos principais benefícios da arquitetura do HEVC Multivista é a fácil implementação sem mudanças na sintaxe ou no processo de descodificação da camada única do HEVC abaixo do nível de cabeçalho da *slice*, o que permite o reuso das implementações de codificador e descodificador HEVC existentes sem grandes alterações para aplicações multivista [13]. Um exemplo da estrutura de previsão do HEVC Multivista é apresentado na Figura 5.1 mostrando as dependências de previsão no tempo e entre vistas. Nesta figura, a vista 0 representa a vista base e, as vistas 1 e 2 são vistas dependentes. Tramas indicadas por "I" usam somente previsão intra-trama; tramas indicadas por "P" utilizam previsão inter-trama uni-preditiva; e tramas "B" utilizam a

predição inter-trama bi-preditiva. As setas horizontais indicam predição temporal, dentro de uma vista e as setas verticais indicam a predição *inter-vista* entre as diferentes vistas.

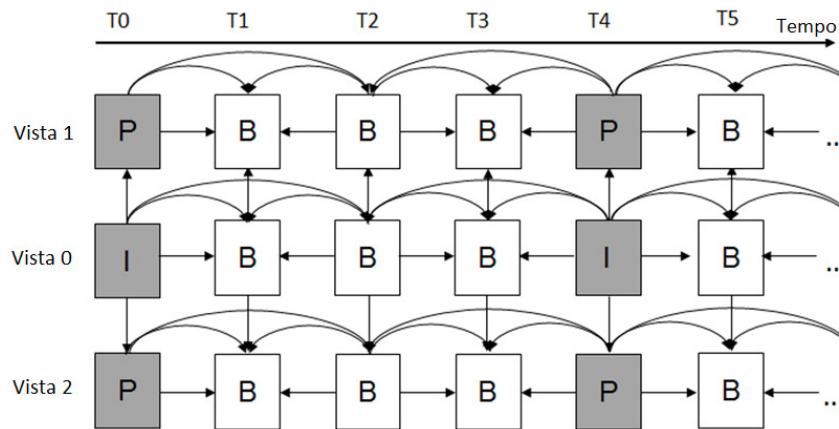


Figura 5.1: Exemplo da estrutura de predição do HEVC Multivista.

Considerando a configuração *Random Access* (RA), descrita na Seção 2.4.3.3, para codificação de três vistas, o HEVC Multivista proporciona uma economia média da taxa de bits de 38% em relação ao HEVC *Simulcast* [13], resultante da predição inter-vistas de textura. Essa predição inter-vista explora a correlação entre as vistas de textura através das otimizações na estrutura de lista de tramas de referência, de modo que as tramas decodificadas a partir de outras vistas podem ser usadas como referência para a predição. Além disso, as funções da predição com compensação de movimento são reusadas para realizar a predição com compensação de disparidade [13][90].

No entanto, considerando a configuração *all-intra* adotada neste trabalho, a codificação *all-intra* no HEVC Multivista é semelhante à codificação *simulcast all-intra* do HEVC, com a diferença de que todas as vistas serão encapsuladas em *access units* de um único *bitstream*. Assim, exceto por esta diferença de encapsulamento, do ponto de vista de operações de codificação, utilizar o software HM (HEVC) para codificar cada vista independentemente quando a configuração *all-intra* é adotada, é equivalente a utilizar o HEVC Multivista com a configuração *all-intra*.

Para comparar a codificação de vídeo multivista utilizando o HEVC *Simulcast* e o HEVC Multivista quando a configuração *all-intra* é adotada, um grupo de experiências foi realizado utilizando as sequências de teste apresentadas na Tabela 4.1, com três vistas cada uma [86]. Estas experiências foram realizadas utilizando o software de referência HTM12.1 da extensão HEVC Multivista [91] e o HEVC *Simulcast* (software de referência

HM10 original) ambos utilizando a configuração *all-intra*. A Tabela 5.1 apresenta os resultados desta comparação, na qual os valores de taxa de bits e PSNR são os valores médios das três vistas codificadas. Em relação aos tempos de codificação listados, no caso do HEVC *Simulcast* os valores reportados são as somas dos tempos de codificação das três vistas, enquanto que, para o HEVC *Multivista* os valores apresentados são os tempos de codificação totais.

Tabela 5.1: Comparação entre HEVC *Simulcast* e HEVC *Multivista*.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	-5,556	0,00	0,83	0,67	-0,03
	30	-0,005	0,00	1,09		
	35	-0,014	0,00	0,64		
	40	-0,022	0,00	-0,57		
<i>Newspaper</i>	25	-0,004	0,00	1,69	-0,73	0,04
	30	-0,003	0,00	1,23		
	35	-0,011	0,00	0,84		
	40	3,658	-0,16	0,61		
<i>Kendo</i>	25	-0,010	0,00	0,89	0,02	-0,001
	30	-0,007	0,00	0,20		
	35	-0,021	0,00	-0,12		
	40	-0,033	0,00	1,45		
<i>GTFly</i>	25	-0,002	0,00	0,99	0,01	-0,0002
	30	-0,002	0,00	0,91		
	35	-0,007	0,00	0,65		
	40	-0,014	0,00	0,78		
<i>PoznanStreet</i>	25	-0,002	0,00	4,52	0,004	-0,0001
	30	-0,002	0,00	1,93		
	35	-0,005	0,00	0,65		
	40	-0,010	0,00	0,87		
<i>PoznanHall2</i>	25	-0,007	0,00	-0,12	0,02	-0,0004
	30	-0,006	0,00	0,30		
	35	-0,020	0,00	0,06		
	40	-0,035	0,00	1,49		
Média		-0,07	-0,007	0,68	0,0010	0,0014

Os resultados mostram que em termos de eficiência de codificação os resultados do HEVC *Simulcast* e HEVC *Multivista* são muito semelhantes, apresentando

valores de Bjontegaard deltas [40][41] próximos de zero. Além disso, considerando o tempo de processamento, os resultados também são muito semelhantes, com o HEVC *Simulcast* apresentando uma redução no tempo de codificação de 0,68% em média, em relação ao HEVC Multivista.

Esta comparação confirmou o que era esperado a partir das especificações do HEVC e HEVC Multivista, nomeadamente, que para o caso *all-intra* considerando as operações de codificação, sem o empacotamento NAL, o HEVC Multivista é funcionalmente idêntico ao HEVC *Simulcast*. Portanto, para simplificar as primeiras avaliações experimentais usando múltiplas vistas, os algoritmos rápidos para predição inter-vista do modo intra e predição inter-vista da profundidade da árvore de codificação propostos nesta seção utilizaram o HEVC *Simulcast* (HM10) com configuração *all-intra* durante a codificação das vistas.

5.1.1. Predição Inter-Vista do Modo Intra

O método rápido de predição inter-vista do modo intra apresentado nesta seção explora a correlação entre os modos escolhidos no processo de codificação intra de vistas vizinhas para reduzir a complexidade da codificação de vídeo multivista. Neste método, a escolha do modo a ser utilizado na codificação de uma PU de uma vista é realizada com base nos modos já utilizados na codificação intra das PUs co-localizadas em vistas vizinhas.

Para iniciar essa exploração, a correlação existente entre os modos intra de vistas adjacentes foi analisada através de experiências realizadas utilizando as sequências de vídeo listadas na Tabela 4.1, com a codificação intra independente das Vistas 0, 1 e 2. Estas experiências foram realizadas utilizando o software de referência HM10 com os QPs 25, 30, 35 e 40.

Os modos intra selecionados nas PUs da Vista 0 (vista par ou vista à esquerda), Vista 1 (vista ímpar ou vista central) e Vista 2 (vista par ou vista à direita) foram analisados, e os casos em que o mesmo modo intra foi selecionado em PUs co-localizadas de diferentes vistas foram contados, considerando cada modo intra individualmente. Além disso, os modos intra escolhidos nas PUs à esquerda e acima das PUs co-localizadas nas Vistas 0 e 2 também foram examinados para verificar se o modo selecionado na PU da Vista 1, era igual a algum dos modos usados nas PUs das Vistas 0 ou 2.

Conforme se pode verificar na coluna "Vista Esquerda - Igual" da Tabela 5.2, aproximadamente 79% dos modos escolhidos na Vista 1 (vista central) eram iguais aos modos selecionados nas PUs co-localizadas da Vista 0, ou nas PUs à esquerda e acima das PUs co-localizadas da Vista 0. O mesmo comportamento foi observado quando comparamos os modos intra escolhidos na Vista 1 com os modos intra selecionados na Vista 2, conforme apresentado na coluna "Vista Direita - Igual" da Tabela 5.2. Estes resultados confirmaram a existência de uma correlação significativa entre os modos de codificação intra escolhidos em vistas vizinhas.

Tabela 5.2: Modos intra da vista ímpar em relação aos modos intra das vistas pares.

Sequência	QP	Vista Esquerda (Vista 0)		Vista Direita (Vista 2)	
		Dif.	Igual	Dif.	Igual
<i>Balloons</i>	25	12,69	87,31	11,22	88,78
	30	10,22	89,78	8,81	91,19
	35	16,65	83,35	19,11	80,89
	40	22,40	77,60	19,34	80,66
<i>Newspaper</i>	25	15,34	84,66	18,60	81,40
	30	27,40	72,60	18,97	81,03
	35	20,67	79,33	18,28	81,72
	40	15,05	84,95	15,05	84,95
<i>Kendo</i>	25	15,34	84,66	18,60	81,40
	30	27,40	72,60	18,97	81,03
	35	20,67	79,33	18,28	81,72
	40	15,05	84,95	15,05	84,95
<i>GTFly</i>	25	14,83	85,17	11,82	88,18
	30	15,65	84,35	13,38	86,62
	35	22,58	77,42	12,48	87,53
	40	13,65	86,35	12,47	87,53
<i>PoznanStreet</i>	25	24,86	75,14	19,60	80,40
	30	33,18	66,82	27,90	72,10
	35	19,29	80,71	27,07	72,93
	40	29,24	70,76	11,95	88,05
<i>UndoDancer</i>	25	21,87	78,13	23,70	76,30
	30	30,00	70,00	24,47	75,53
	35	28,20	71,80	30,65	69,35
	40	26,38	73,62	29,36	70,64
Média		20,78	79,22	18,55	81,45

Com base nas análises apresentadas acima, foi desenvolvido um método rápido de decisão de modo intra com inferência inter-vistas. Os detalhes do método proposto são apresentados a seguir.

A Figura 5.2 apresenta de forma simplificada o processo de codificação utilizado para implementar a solução proposta, no qual após a codificação das vistas pares (Vista 0 e Vista 2, neste exemplo) ser efetuada utilizando o modo de decisão intra rápido apresentado na Seção 4.1.3, a vista ímpar (Vista 1, neste exemplo) é codificada utilizando o método de predição inter-vistas proposto nesta seção, de forma que as informações sobre os modos intra selecionados nas vistas pares são usados para acelerar a codificação da vista ímpar.

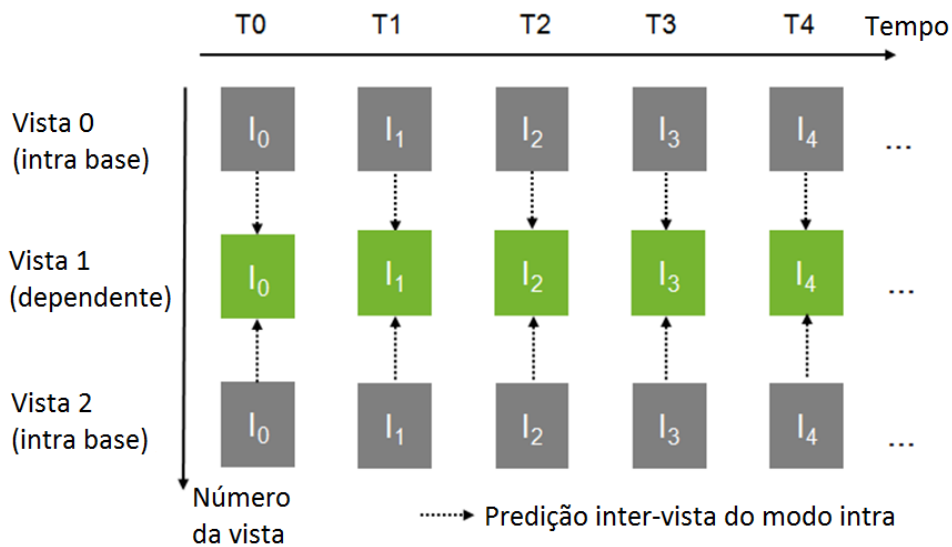


Figura 5.2: Codificação *all-intra* usando o esquema proposto para codificação HEVC baseada em múltiplas vistas.

Uma descrição simplificada das operações e do fluxo de informação do novo método de predição de modo inter-vistas são representados graficamente na Figura 5.3.

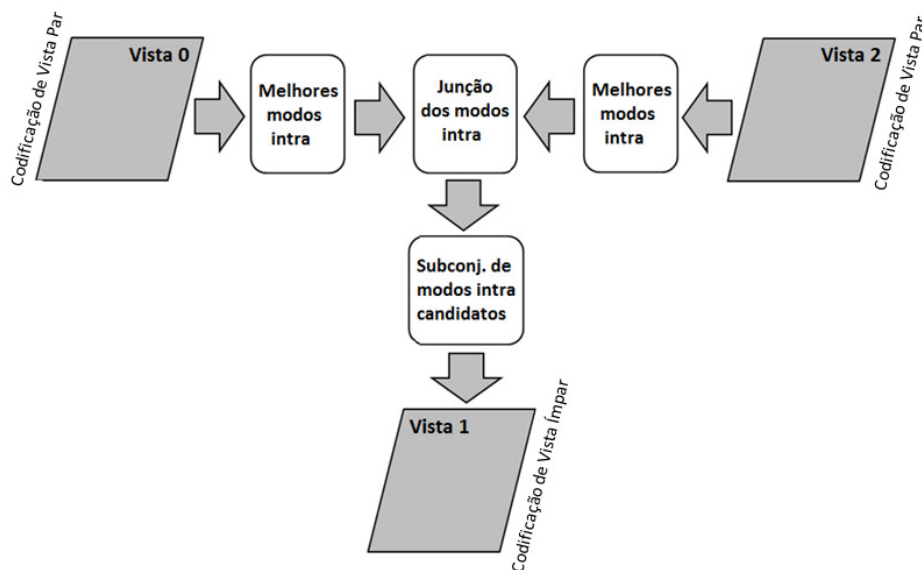


Figura 5.3: Predição inter-vista do modo intra proposto para a extensão multivista do HEVC.

Inicialmente, as Vistas 0 e 2 são codificadas utilizando o algoritmo rápido de decisão de modo intra apresentado na Seção 4.1.3. Logo após, os modos intra escolhidos para codificar estas vistas são agrupados para compor o subconjunto de modos candidatos que serão utilizados para codificar a Vista 1, de forma que durante a codificação intra das PUs co-localizadas (no tempo e espaço) da Vista 1, é efetuada uma predição inter-vista do modo intra, utilizando como base para a predição os melhores modos intra selecionados durante a codificação das vistas pares.

Com este procedimento de predição inter-vista, o processo de escolha do subconjunto de modos intra candidatos que é realizado quando codificamos as Vistas 0 e 2, não é mais necessário para a codificação da Vista 1, e de forma mais geral, já não é mais necessário para a codificação das vistas ímpares. Um esquema mais detalhado do algoritmo rápido proposto é apresentado no fluxograma da Figura 5.4.

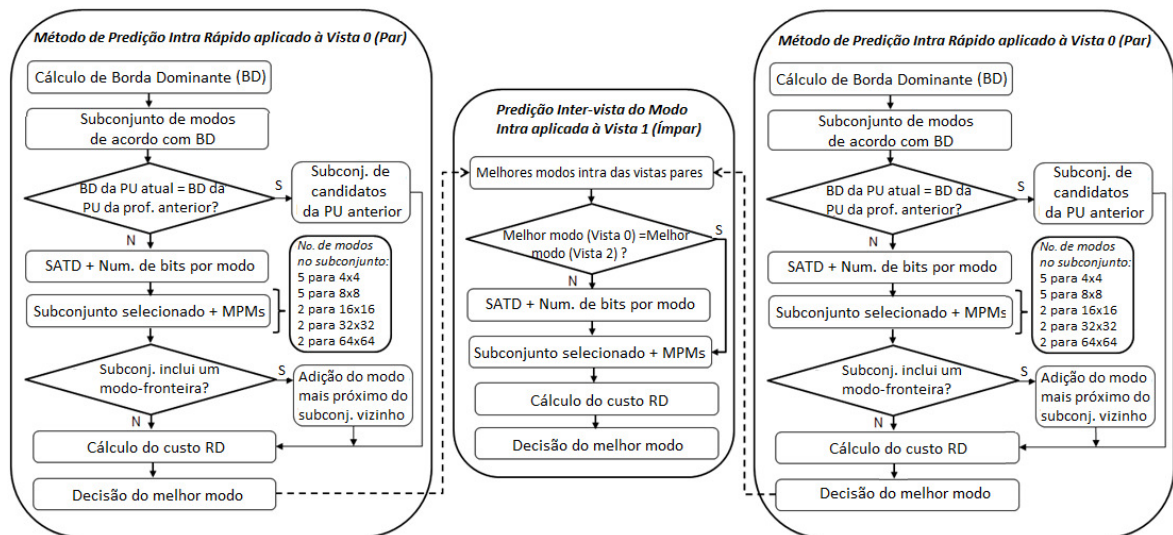


Figura 5.4 – Fluxograma do método de predição inter-vista do modo intra proposto para a codificação de vídeo multivista baseada no codificador HEVC.

Para avaliar a redução de complexidade obtida utilizando o algoritmo proposto e o seu efeito no desempenho da codificação, um conjunto de experiências foi realizado com uma versão modificada do software de referência HM10, modificado com a inclusão do método descrito nesta seção. As avaliações experimentais utilizaram as seqüências de teste listadas na Tabela 4.1, as quais foram codificadas os QPs 25, 30, 35 e 40, usando a configuração *all-intra*.

A Tabela 5.3 apresenta os resultados detalhados para $\Delta PSNR$, $\Delta Taxa de Bits$, $\Delta Tempo$ e Bjontegaard deltas, comparando o desempenho e a complexidade do algoritmo proposto em relação ao HEVC *Simulcast* (software de referência HM10).

É importante observar que nestes testes foram codificadas as três vistas e os ganhos deste novo método de predição inter-vistas são calculados para a vista central, uma vez que as outras vistas foram completamente codificadas utilizando o algoritmo rápido para predição intra apresentado na Seção 4.1.3, sem utilizar informações das vistas vizinhas.

Tabela 5.3: Comparação do método de predição inter-vista do modo intra em relação ao HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 2.

Sequência	QP	$\Delta Taxa de Bits$ [%]	$\Delta PSNR$ [dB]	$\Delta Tempo$ [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	0,93	-0,04	-44,25	1,8	-0,10
	30	1,08	-0,04	-46,21		
	35	1,12	-0,04	-45,19		
	40	1,22	-0,05	-45,87		
<i>Newspaper</i>	25	0,89	-0,03	-43,70	1,8	-0,10
	30	1,01	-0,03	-45,71		
	35	1,28	-0,04	-44,09		
	40	1,41	-0,04	-45,87		
<i>Kendo</i>	25	0,81	-0,04	-44,30	1,9	-0,09
	30	0,93	-0,04	-43,53		
	35	1,02	-0,05	-43,39		
	40	1,21	-0,05	-45,23		
<i>GTFly</i>	25	0,95	-0,03	-44,62	2,0	-0,07
	30	1,07	-0,04	-45,43		
	35	1,12	-0,04	-42,69		
	40	1,21	-0,04	-44,23		
<i>PoznanStreet</i>	25	0,82	-0,03	-46,35	1,9	-0,07
	30	0,97	-0,04	-45,73		
	35	1,02	-0,04	-47,35		
	40	1,30	-0,03	-45,83		
<i>UndoDancer</i>	25	0,85	-0,03	-47,69	1,8	-0,07
	30	0,95	-0,03	-48,81		
	35	1,08	-0,03	-48,32		
	40	1,22	-0,03	-47,77		
Média		1,06	0,038	-45,51	1,87	-0,08

De acordo com os resultados das simulações, o algoritmo proposto nesta seção alcançou uma redução média no tempo de codificação de 45,51% em relação ao HEVC *Simulcast*, com uma degradação mínima em eficiência de codificação de 0,08dB em média, considerando o BD-PSNR. Isto significa que a solução proposta é capaz de reduzir o tempo de processamento em pelo menos 6,39%, quando comparado com o uso da predição intra rápida apresentada na Seção 4.1.3, para cada uma das três vistas.

Uma vez que o desempenho da predição inter-vista depende naturalmente da semelhança visual entre as vistas vizinhas, as quais por sua vez são afetadas por fatores externos às operações de codificação, tais como o arranjo das câmeras e distância entre os objetos da cena e as câmeras, é importante avaliar o efeito destes fatores no funcionamento da solução proposta. No caso da distância entre os objetos e a câmera, e uma vez que quanto menor é a distância dos objetos até a câmera, maior é a disparidade, é de esperar que a predição inter-vista do modo intra seja menos eficaz para PUs correspondentes a áreas da trama que representem objetos próximos.

Para verificar o efeito do aumento da distância entre vistas no desempenho do método de predição de modo, algumas experiências foram realizadas para comparar os resultados do método proposto em relação ao HEVC *Simulcast*. Estes testes foram realizados sobre as sequências de vídeo especificadas em [92], usando a configuração *all-intra*. Detalhes destas sequências são apresentados na Tabela 5.4.

Tabela 5.4: Sequências de teste para experimentos adicionais.

Sequência	Taxa de Tramas (fps)	Número de Tramas	Resolução (pixels)
Ballroom	25	100	640x480
Loverbird1	30	100	640x480
Flamenco2	30	100	640x480
Objects	30	100	640x480
Vassar	25	100	640x480
Crowd	30	100	640x480

No primeiro teste, a Vista 1 (vista dependente) foi codificada utilizando os melhores modos intra das Vistas 0 e 2. Os resultados obtidos são apresentados na Tabela 5.5.

Tabela 5.5: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 2.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNR Y [dB]
<i>Ballroom</i>	22	0,80	-0,03	-42,52	2,2	-0,12
	27	1,49	-0,03	-42,89		
	32	1,69	-0,03	-44,78		
	37	2,14	-0,04	-43,18		
<i>Loverbird1</i>	22	0,72	-0,03	-42,13	1,8	-0,08
	27	0,89	-0,03	-43,26		
	32	1,58	-0,02	-43,58		
	37	1,82	-0,03	-44,05		
<i>Flamenco2</i>	22	1,05	-0,03	-42,67	2,1	-0,13
	27	1,28	-0,04	-43,99		
	32	1,68	-0,04	-45,48		
	37	2,13	-0,04	-46,39		
<i>Vassar</i>	22	0,86	-0,02	-42,04	1,8	-0,12
	27	1,27	-0,02	-41,45		
	32	1,43	-0,03	-43,67		
	37	2,02	-0,03	-44,03		
<i>Objects</i>	22	0,41	-0,02	-43,15	1,6	-0,06
	27	0,71	-0,03	-45,23		
	32	1,17	-0,03	-45,16		
	37	1,92	-0,03	-46,09		
<i>Crowd</i>	22	0,44	-0,02	-41,29	1,4	-0,10
	27	0,68	-0,02	-43,74		
	32	1,55	-0,03	-44,44		
	37	1,89	-0,02	-45,12		
Média		1,317	-0,029	-43,76	1,82	-0,10

Logo após, a Vista 1 foi codificada utilizando os melhores modos intra das Vistas 0 e 3, e depois utilizando os melhores modos intra das Vistas 0 e 4. Os resultados detalhados de eficiência de codificação e complexidade destes dois experimentos são apresentados nas Tabela 5.6 e Tabela 5.7, respectivamente. Tais resultados apresentam comparações de performance e tempo de processamento da Vista 1 codificada usando o método de predição inter-vistas proposto, em relação à codificação da Vista 1 usando o HEVC *Simulcast*.

Tabela 5.6: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 3.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNR Y [dB]
<i>Ballroom</i>	22	0,84	-0,03	-41,19	2,2	-0.12
	27	1,58	-0,03	-41,50		
	32	1,72	-0,03	-43,36		
	37	2,01	-0,05	-45,12		
<i>Loverbird1</i>	22	0,70	-0,02	-42,27	1,8	-0.08
	27	0,79	-0,03	-43,25		
	32	1,51	-0,03	-44,85		
	37	1,88	-0,03	-46,75		
<i>Flamenco2</i>	22	1,10	-0,04	-42,05	2,1	-0.13
	27	1,39	-0,04	-43,89		
	32	1,35	-0,04	-45,16		
	37	2,36	-0,04	-44,17		
<i>Vassar</i>	22	0,80	-0,03	-40,95	1,9	-0.12
	27	1,22	-0,03	-41,87		
	32	1,50	-0,03	-41,99		
	37	2,14	-0,04	-43,67		
<i>Objects</i>	22	0,48	-0,03	-43,45	1,7	-0.07
	27	0,77	-0,03	-44,23		
	32	1,34	-0,03	-44,89		
	37	1,81	-0,03	-46,95		
<i>Crowd</i>	22	0,57	-0,03	-40,75	1,5	-0.10
	27	0,63	-0,03	-42,47		
	32	1,65	-0,03	-42,35		
	37	1,87	-0,03	-44,06		
Média		1,334	-0,033	-43,38	1,87	-0,10

De acordo com os resultados da Tabela 5.5, o algoritmo proposto para a predição inter-vista do modo intra, utilizado para codificar a Vista 1 a partir das Vistas 0 e 2, alcançou uma redução média no tempo de codificação de 43,76% quando comparado com a codificação da Vista 1 utilizando ao HEVC Simulcast, com um aumento em BD-rate de 1,82%, em média. Na Tabela 5.6, são apresentados os resultados do algoritmo proposto, considerando a Vista 1 codificada a partir das Vistas 0 e 3, o qual efetuou uma redução média no tempo de codificação de 43,38% em comparação com a codificação da Vista 1 usando ao HEVC Simulcast, com um aumento em BD-rate de 1,87%, em média. Na

Tabela 5.7, são apresentados os resultados do algoritmo proposto, considerando a Vista 1 codificada a partir das Vistas 0 e 4, apresentou uma redução média no tempo de codificação de 41,19% em relação ao HEVC *Simulcast*, com um aumento em *BD-rate* de 1,88%, em média.

Com base nos resultados apresentados nas Tabela 5.5, Tabela 5.6 e Tabela 5.7, é possível concluir que, conforme a distância entre a vista base e as vistas dependentes aumenta, a eficiência de codificação da predição inter-vista de modo intra diminui, enquanto a redução de complexidade apresenta valores semelhantes.

Tabela 5.7: Comparação do método de predição inter-vista do modo intra proposto versus HEVC Simulcast considerando a Vista 1 codificada a partir das Vistas 0 e 4.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Ballroom</i>	22	0,91	-0,04	-41.83	2,3	-0,12
	27	1,47	-0,03	-41.22		
	32	1,84	-0,04	-43.75		
	37	2,08	-0,05	-45.18		
<i>Loverbird1</i>	22	0,78	-0,03	-42.34	1,9	-0,08
	27	0,84	-0,03	-43.99		
	32	1,65	-0,04	-44.06		
	37	2,08	-0,04	-46.32		
<i>Flamenco2</i>	22	0,96	-0,04	-43.22	2,0	-0,12
	27	1,09	-0,04	-43.79		
	32	1,26	-0,04	-45.13		
	37	2,13	-0,05	-46.72		
<i>Vassar</i>	22	0,83	-0,03	-44.09	1,9	-0,13
	27	1,29	-0,03	-44.88		
	32	1,46	-0,03	-46.89		
	37	2,02	-0,04	-47.02		
<i>Objects</i>	22	0,56	-0,02	-44.78	1,7	-0,07
	27	0,62	-0,03	-45.95		
	32	1,46	-0,03	-46.87		
	37	1,92	-0,03	45.54		
<i>Crowd</i>	22	0,66	-0,03	-45.25	1,5	-0,10
	27	0,78	-0,03	-46.84		
	32	1,47	-0,03	-47.21		
	37	1,71	-0,03	-46.88		
Média		1,331	-0,035	-41,19	1,88	-0,10

A predição inter-vistas dos modos a serem utilizados na codificação intra das vistas vizinhas ainda poderia ser explorada através da compensação do efeito de disparidade inter-vistas. Esta possibilidade foi considerada, mas tendo em vista que a estimação da disparidade diminuiria a redução da complexidade alcançada pelo método de decisão de modo rápido proposto, e a redução de complexidade é o objetivo principal desta solução, decidiu-se não utilizar a compensação de disparidade no método e realizar a predição inter-vista do modo intra com base nos modos utilizados na codificação intra das PUs co-localizadas de vistas vizinhas (conforme explicado na Seção 5.1.1, Tabela 5.2).

Levando em consideração este fato e os resultados dos experimentos de variação da distância entre as vistas, a solução adotada neste trabalho, a qual utiliza os melhores modos intra das Vistas 0 e 2 para codificar a Vista 1, foi a que apresentou o melhor equilíbrio entre a eficiência de codificação e a redução da complexidade.

5.1.2. Predição Inter-Vistas da Profundidade da Árvore de Codificação

Conforme mencionado na Seção 2.4.1, cada CTU pode ser composta por uma CU ou dividida em múltiplas CUs utilizando uma estrutura de *quadtree* [93]. Cada CU-folha na árvore de codificação pode ser dividida em duas ou quatro PUs. Além disso, cada CU ainda é raiz de outra estrutura em árvore, a *quadtree* residual (RQT), na qual as folhas são TUs. Considerando-se que a árvore RQT está enraizada nas CUs-folha da árvore de codificação, uma *quadtree* aninhada é formada.

Para melhorar a eficiência de codificação, o codificador HEVC define a melhor segmentação da estrutura de *quadtree* utilizando técnicas de otimização da taxa-distorção (RDO), as quais testam todas as partições possíveis da árvore de codificação para encontrar aquela com o menor custo da taxa-distorção (RD).

No entanto, este processo de pesquisa e decisão do melhor particionamento dos blocos requer uma grande complexidade computacional no codificador, motivando o desenvolvimento de diversas soluções que pretendem reduzir esta complexidade através de algoritmos de terminação antecipada ou "poda" da árvore de codificação. Além disso, tendo em vista que o particionamento e profundidade da árvore de CUs influencia na complexidade da codificação intra, a qual é o foco principal desse trabalho, soluções para otimização das decisões de profundidade da árvore de codificação também são propostas neste capítulo.

Conforme o exemplo de árvore de codificação da Figura 2.3, rerepresentada na Figura 5.5 para facilitar a leitura do texto, é possível observar que o número de diferentes árvores a serem testadas no processo RDO aumenta rapidamente (exponencialmente com base 4) de acordo com a profundidade máxima da árvore.

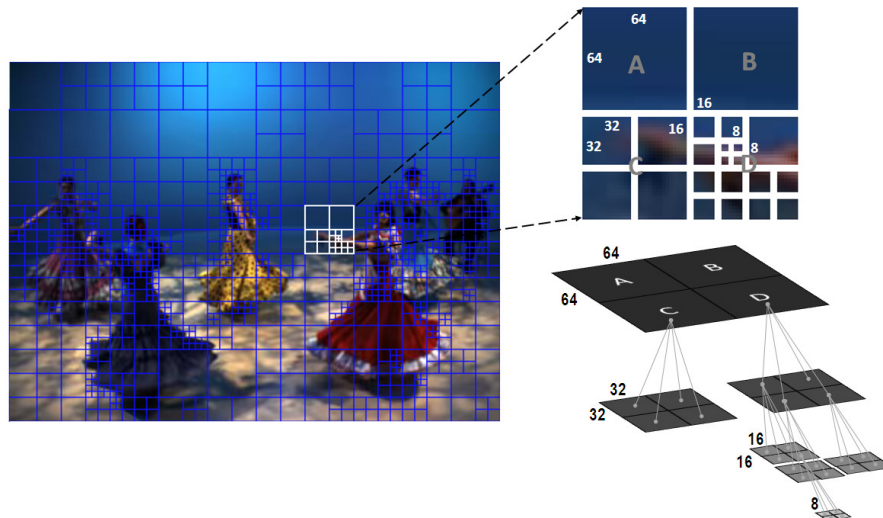


Figura 5.5: Exemplo de estrutura de codificação usando quadrees.

Recentemente, alguns trabalhos visando a redução de complexidade da árvore de codificação do HEVC têm sido publicados. Em [94], um algoritmo rápido para decisão da profundidade da árvore de CUs é proposto com base nas correlações de profundidade espaço-temporais ao nível de sub-CU. Em [95], um método rápido de decisão da profundidade da CU é proposto através da redução no intervalo de pesquisa da profundidade. Em [96], um método de controlo de complexidade é proposto, o qual reduz o número de possibilidades de árvore de codificação avaliadas através do RDO, limitando a profundidade máxima de acordo com a CTU co-localizada na trama anterior. Em [97], um algoritmo de decisão da profundidade das CUs para codificação inter-trama usando uma análise das informações de profundidade da CTU é apresentado. Em [98], ferramentas de *data mining* são usadas para construir um conjunto de árvores de decisão que permitem a terminação antecipada dos processos de decisão quando a melhor configuração para a árvore de codificação é encontrada. Um algoritmo rápido baseado na profundidade da unidade de codificação (CU), o qual prevê a estrutura de divisão da CU atual utilizando a mesma estrutura da CU co-localizada na trama de referência ou temporalmente anterior é proposto em [99]. A divisão da CU com terminação antecipada para a codificação inter-trama é proposta em [100]. Em [101], apresenta-se um método de decisão do tamanho da

CU através da terminação antecipada do processo de estimação de movimento desnecessário em CUs de tamanho menor. Estes métodos pretendem explorar as correlações espaciais e/ou temporais, mas a correlação inter-vistas das decisões de profundidade da árvore de CUs em vistas adjacentes ainda é pouco explorada na codificação multivista.

Nesta seção, um método para a predição inter-vistas das decisões de profundidade da árvore de CUs é proposto para a extensão multivista do HEVC, explorando a relação entre as decisões de profundidade das CU em vistas vizinhas.

O método proposto nesta seção é representado na Figura 5.6, na qual as seguintes etapas do processo de codificação são apresentadas. Inicialmente, a vista base (Vista 0 na Figura 5.6) é codificada utilizando as novas técnicas adotadas no HEVC, tais como: estrutura de *quadtree* hierárquica, particionamento flexível de bloco, maior número de direções de predição, blocos de tamanho maior, etc. Na codificação da vista base, as estruturas de profundidade da árvore de CUs foram codificadas usando o processo convencional utilizado no HM10, no qual todas as partições possíveis da árvore de codificação são testadas para encontrar aquela com menor custo RD. Logo após, as decisões de profundidade máxima das CUs codificadas na vista base foram usadas para auxiliar na codificação das vistas dependentes, conforme ilustrado na Figura 5.6. Assim, as profundidades máximas das CUs definidas durante a codificação das CTUs da vista base foram usadas como limites de profundidade das árvores de CUs na codificação das CTUs das vistas dependentes, conforme apresentado no fluxograma da Figura 5.7.

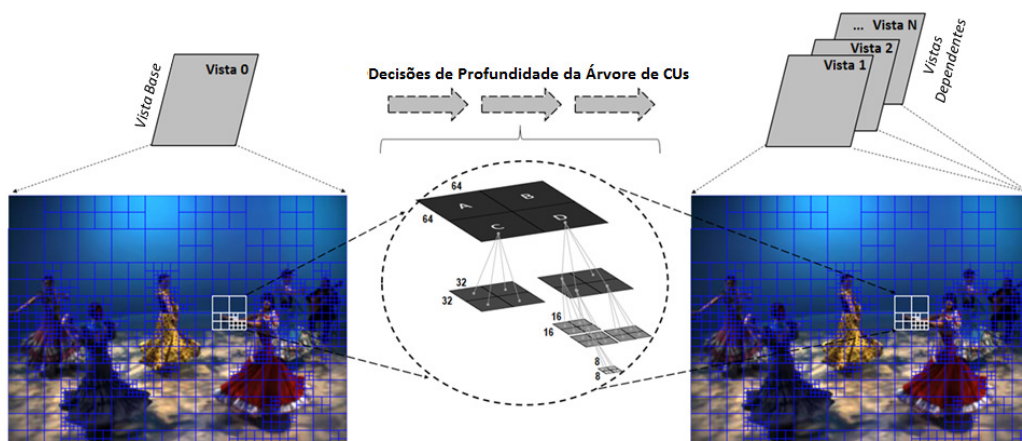


Figura 5.6: Predição inter-vista da profundidade da árvore de codificação para a extensão multivista do HEVC.

De modo que, se a profundidade da CU atual da vista dependente é menor do que a profundidade máxima da CU pertencente a CTU correspondente na vista base, a CU

atual é dividida em quatro sub-CUs iguais, caso contrário, a próxima CTU da vista dependente é codificada. Este método de predição inter-vistas da estrutura da árvore de CUs foi desenvolvido para acelerar o processo de codificação multivista, reduzindo a complexidade computacional, porém mantendo a performance de codificação.

Desta forma, durante a codificação das vistas dependentes deixa de ser necessário executar de forma exaustiva o processo de pesquisa em busca da melhor segmentação da estrutura de *quadtree*.

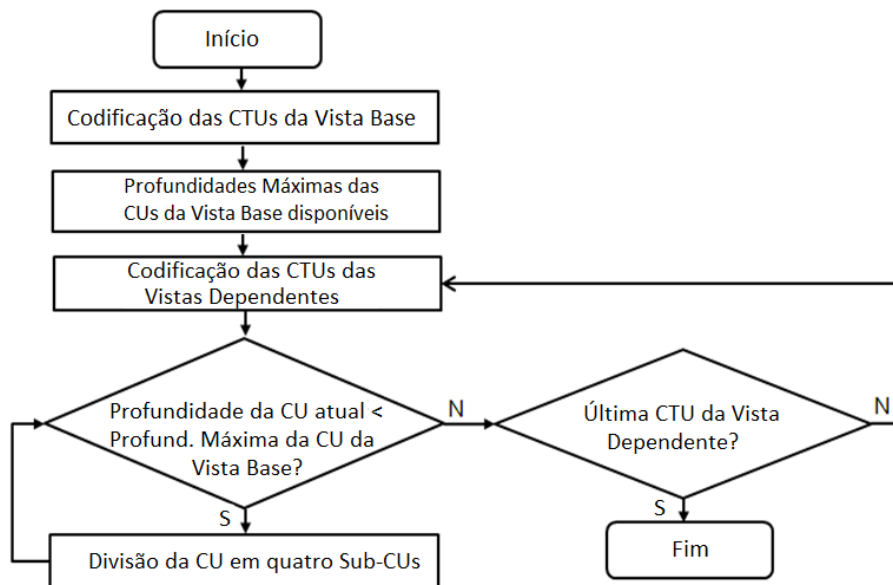


Figura 5.7: Fluxograma do método de predição inter-vistas da profundidade da árvore de codificação.

Para avaliar a redução da complexidade obtida utilizando o algoritmo proposto e o seu efeito no desempenho da codificação, um grupo de experimentos foi realizado utilizando o software de referência HM10, o qual foi modificado para incluir o método descrito nesta seção. Os experimentos foram conduzidos utilizando as sequências de teste apresentadas da Tabela 4.1, conforme especificado em [86].

A Tabela 5.8 apresenta os resultados da codificação da Vista 1 comparando o desempenho e a complexidade quando o método de predição inter-vistas proposto é utilizado, em relação a codificação da Vista 1 usando o HEVC *Simulcast*.

Tabela 5.8: Comparação do método de predição inter-vistas proposto em relação ao HEVC Simulcast considerando a codificação da Vista 1.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNR Y [dB]
<i>Balloons</i>	25	0,19	-0,01	-15,97	0,6	-0,03
	30	0,07	-0,05	-22,52		
	35	0,25	-0,05	-29,20		
	40	0,42	-0,13	-37,51		
<i>Newspaper</i>	25	-0,07	-0,02	-12,20	0,3	-0,01
	30	0,02	-0,02	-22,78		
	35	0,01	-0,04	-28,11		
	40	0,21	-0,02	-29,17		
<i>Kendo</i>	25	0,11	-0,01	-34,65	0,7	-0,04
	30	0,17	-0,04	-38,77		
	35	0,38	-0,05	-43,02		
	40	0,51	-0,11	-48,80		
<i>GTFly</i>	25	-0,17	-0,02	-23,06	0,2	-0,01
	30	0,07	0,00	-37,00		
	35	0,06	-0,02	-42,90		
	40	0,07	-0,06	-50,32		
<i>PoznanStreet</i>	25	0,05	-0,01	-18,18	0,5	-0,02
	30	-0,02	-0,03	-20,10		
	35	0,09	-0,05	-27,99		
	40	0,45	-0,10	-38,05		
<i>UndoDancer</i>	25	-0,07	-0,02	-12,20	0,9	-0,05
	30	0,02	-0,02	-22,78		
	35	0,01	-0,04	-28,11		
	40	0,21	-0,02	-29,17		
Média		0,13	-0,039	-29,69	0,53	-0,03

De acordo com os resultados das simulações apresentados na Tabela 5.8, o método proposto neste trabalho permitiu alcançar uma redução no tempo de codificação de até 50,32% em relação ao HEVC *Simulcast*, e cerca de 29% em média, com uma degradação muito pequena em termos de eficiência de codificação (0,03dB em BD-PSNR, em média).

A Tabela 5.9 apresenta os resultados da codificação da Vista 2 utilizando o método proposto em comparação com os da codificação da Vista 2 usando o HEVC

Simulcast. A partir destes resultados, é possível verificar que o método proposto conseguiu uma redução no tempo de codificação de até 49,51% (29,05% em média) com uma degradação mínima na performance de codificação (0,03dB em BD-PSNR, em média).

Tabela 5.9: Comparação do método de predição inter-vistas proposto em relação ao HEVC Simulcast considerando a codificação da Vista 2.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	0,31	-0,02	-15,31	1,0	-0,06
	30	0,41	-0,04	-21,64		
	35	0,54	-0,10	-28,15		
	40	0,68	-0,20	-38,35		
<i>Newspaper</i>	25	0,65	-0,04	-12,09	1,4	-0,07
	30	0,59	-0,06	-14,32		
	35	0,94	-0,10	-21,81		
	40	1,18	-0,10	-30,01		
<i>Kendo</i>	25	0,03	-0,001	-34,43	0,1	-0,004
	30	-0,07	0,0005	-38,89		
	35	-0,03	-0,03	-43,35		
	40	0,06	-0,04	-48,58		
<i>GTFLy</i>	25	-0,21	-0,03	-22,81	0,4	-0,01
	30	0,09	-0,01	-36,27		
	35	0,16	-0,03	-43,00		
	40	0,12	-0,07	-49,51		
<i>PoznanStreet</i>	25	0,05	-0,01	-18,18	0,5	-0,02
	30	-0,02	-0,03	-20,10		
	35	0,09	-0,05	-27,99		
	40	0,45	-0,10	-38,05		
<i>UndoDancer</i>	25	-0,01	-0,03	-14,25	0,6	-0,02
	30	0,16	-0,03	-22,60		
	35	0,16	-0,06	-28,31		
	40	0,54	-0,04	-29,18		
Média		0,29	-0,051	-29,05	0,67	-0,03

Esta seção apresentou um método para a predição inter-vistas da estrutura da árvore de codificação, a qual utiliza as profundidades máximas das árvores de CU definidas durante a codificação independente da vista base, como um limiar a ser utilizado

na codificação das vistas dependentes. Assim, as comparações do custo de taxa-distorção (RD) de todas as divisões de CU possíveis não são mais necessárias para codificar essas vistas dependentes, reduzindo assim a complexidade de codificação.

Os resultados experimentais demonstraram que quando comparado com o HEVC *Simulcast*, o tempo de processamento da codificação foi reduzido em cerca de 29% em média, com uma perda negligenciável em eficiência de codificação. Tanto quanto sabemos, à data de desenvolvimento deste método, ele era o único a propor uma predição inter-vistas da estrutura de profundidade da árvore de codificação para a redução de complexidade da extensão multivista do HEVC.

5.2. Algoritmos de Aceleração para o Codificador 3D-HEVC

Nesta seção serão apresentados algoritmos utilizando o software de referência HTM 12.1 do 3D-HEVC, o qual pode ser configurado para codificar apenas vistas de textura sem mapas de profundidade, ou codificar vistas de textura juntamente com seus respectivos mapas de profundidade.

5.2.1. Otimização da Decisão de Profundidade Máxima da Árvore de CUs

A extensão multivista do HEVC, visa tirar proveito da redundância entre as diferentes vistas de textura da mesma cena capturadas a partir de pontos de perspectiva ligeiramente diferentes, redefinindo as funções de predição do movimento compensado existentes para realizar a predição inter-vistas da disparidade compensada [90].

Nesta seção, um algoritmo de predição inter-vista da profundidade máxima da CU foi desenvolvido para reduzir a complexidade de codificação da extensão 3D-HEVC para vistas de textura, explorando as semelhanças entre as estruturas de profundidade da árvore de CUs em vistas adjacentes.

Para desenvolver esse algoritmo, inicialmente procedeu-se a uma análise estatística utilizando as sequências de teste: ‘PoznanHall2’, ‘Shark’, ‘PoznanCarPark’ e ‘PoznanHall1’ com resolução de 1920x1088 pixels, especificadas em [86] e [102]. As simulações foram realizadas com três vistas por sequência utilizando os QPs: 25, 30, 35 e 40. Os resultados da codificação e padrões de uso da estrutura de *quadtree* foram

analisados, para identificar eventuais correlações entre as decisões de profundidade das CUs definidas durante a codificação de vistas vizinhas.

Os resultados experimentais que são apresentados na Tabela 5.10 mostram as frequências de co-ocorrências das decisões de profundidade máxima das CUs da Vista 2 em relação à profundidade máxima das CUs das Vistas 0 e 1. Na tabela, a primeira coluna apresenta os valores de QP utilizados. A segunda coluna apresenta a percentagem de casos em que a profundidade máxima da CU da Vista 2 é igual apenas à profundidade máxima da CU co-localizada na Vista 0. Da mesma forma a terceira coluna representa a percentagem de casos em que a profundidade máxima da CU da Vista 2 é igual apenas à profundidade máxima da CU da Vista 1. As duas últimas colunas mostram os casos em que a profundidade máxima da CU da Vista 2 é igual a ambas Vistas 0 e 1 ($V_0=V_1=V_2$) e, o caso em que a profundidade máxima da CU da Vista 2 é diferente de ambas as profundidades máximas das CUs das Vistas 0 e 1 ($(V_0 \text{ e } V_1) \neq V_2$), respectivamente.

Tabela 5.10: Análise de co-ocorrências da profundidade máxima das CUs da Vista 2 em relação a profundidade máxima das CUs das vistas 0 e 1.

QP	Vista 0 = Vista 2 apenas [%]	Vista 1 = Vista 2 apenas [%]	Vista 0 & Vista 1 = Vista 2 [%]	Vista 0 & Vista 1 ≠ Vista 2 [%]
25	17,22	28,67	20,78	33,33
30	11,07	31,14	21,50	36,29
35	11,63	34,53	16,63	37,21
40	9,89	32,11	22,80	35,20
Méd.	12,45	31,61	20,43	35,51

Os resultados da Tabela 5.10 foram utilizados como base para a implementação do algoritmo de predição inter-vistas da profundidade da CU proposto nesta seção. De modo que, os casos em que a profundidade máxima das CUs das Vistas 0 e 1 é igual a profundidade máxima da CU da Vista 2 (20,43% na Tabela 5.10) compõem o primeiro teste do algoritmo apresentado no fluxograma da Figura 5.8, como será explicado a seguir. O segundo teste do algoritmo diz respeito aos casos em que a profundidade máxima das CUs das Vistas 0 e 1 é diferente da profundidade máxima da CU da Vista 2 (35,51% na Tabela 5.10).

Além disso, os casos em que a profundidade máxima da CU da Vista 2 é somente igual a da Vista 1 (31,61% na Tabela 5.10) são adicionados ao algoritmo como um passo resultante do segundo teste do algoritmo, quando este teste é verdadeiro. Caso

contrário, se este teste for falso, a profundidade máxima da CU da Vista 2 é definida usando o método padrão adotado no HTM (software de referência do 3D-HEVC) [91]. Assim, o método proposto abrange um total de 87,55% dos casos avaliados na Tabela 5.10.

Os 12,45% restantes correspondem aos casos em que a profundidade máxima da CU da Vista 2 é somente igual a da Vista 0. Estes casos não foram adicionados ao algoritmo, uma vez que sua percentagem de ocorrência é menor do que os casos em que a Vista 2 é somente igual a Vista 1. Assim, o caso com maior percentagem de ocorrência entre esses dois foi escolhido para ser explorado neste algoritmo.

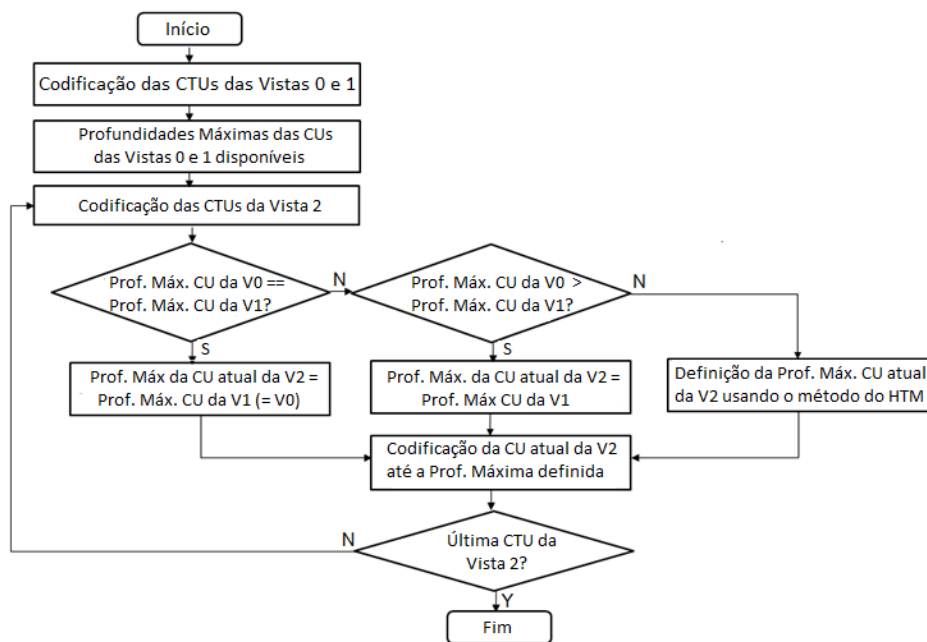


Figura 5.8: Fluxograma do algoritmo proposto para otimização inter-vistas das decisões de profundidade máxima da árvore de CUs.

A Figura 5.9 ilustra o método proposto, no qual os seguintes passos do processo de codificação são realizados. Inicialmente, as Vistas 0 e 1 são codificadas utilizando as ferramentas do 3D-HEVC [11]. Estas duas vistas iniciais são codificadas sem qualquer restrição. Logo após, as decisões de profundidade da árvore de codificação, as quais contêm as profundidades máximas das CUs codificadas nestas vistas, são usadas para acelerar o processo de codificação de uma terceira vista (Vista 2). Assim, os níveis de profundidade máxima das CUs que compõem as CTUs das Vistas 0 e 1 foram utilizados para definir os limites de profundidade das árvores de CU quando as CTUs da Vista 2 são codificadas.

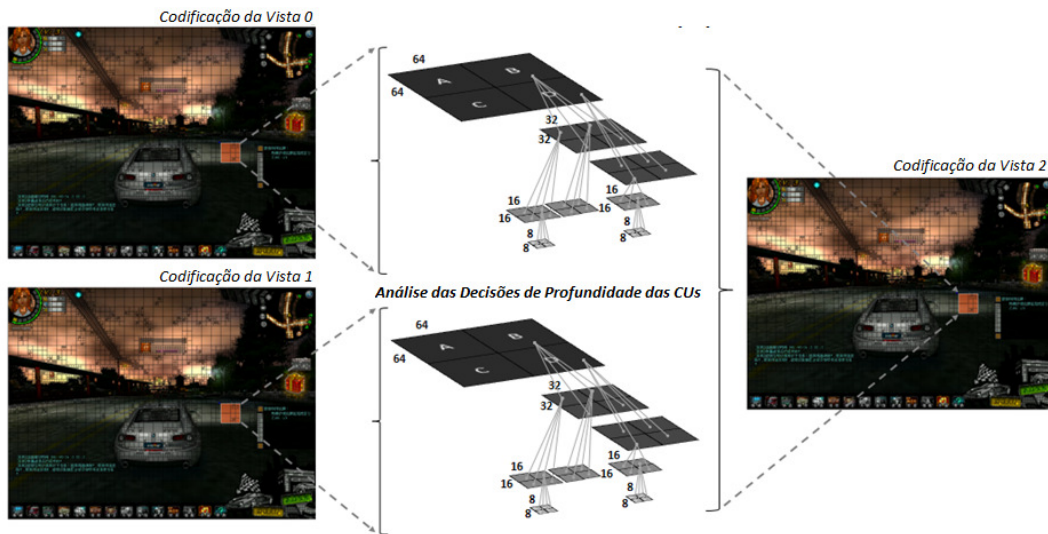


Figura 5.9: Algoritmo proposto para a predição inter-vistas das decisões de profundidade da árvore de CUs do 3D-HEVC.

De acordo com o fluxograma da Figura 5.8, se a profundidade máxima da CU da Vista 0 é igual a profundidade máxima da CU da Vista 1, então a profundidade máxima da CU da Vista 2 é definida como a mesma de ambas as Vistas 0 e 1.

No entanto, se a profundidade máxima da CU da Vista 0 é maior do que a da Vista 1, então a profundidade máxima da CU da Vista 2 é definida como sendo a mesma profundidade máxima da CU da Vista 1. Esta definição foi feita com base no percentual de casos em que a profundidade máxima da CU da Vista 2 é somente igual à Vista 1 (31,61% na Tabela 5.10). Nos outros casos, a profundidade máxima da CU da Vista 2 é definida usando o método padrão de codificação da árvore de CUs, conforme explicado na Seção 5.1.2.

A partir da utilização do algoritmo para a predição inter-vistas da decisão de profundidade máxima da árvore de CUs, o processo de escolha da melhor segmentação da estrutura de *quadtree* com base no custo de RD pode ser suprimido da codificação da Vista 2. Isso causa uma importante redução da complexidade com uma degradação mínima em eficiência de codificação, conforme apresentado nos resultados a seguir.

Para avaliar a redução da complexidade computacional e o desempenho do algoritmo proposto, um grupo de experimentos foi realizado com a versão 12.1 do HTM [91], o qual foi modificado para incluir o método descrito nesta seção. Os experimentos foram conduzidos usando as sequências de teste apresentadas na Tabela 4.1. Essas sequências foram codificadas usando os QPs 25, 30, 35 e 40 [86], com três vistas por sequência.

A Tabela 5.11 apresenta os resultados de codificação da Vista 2, comparando o método de predição inter-vistas da profundidade da árvore de CU proposto em relação à codificação da Vista 2 usando o 3D-HEVC sem alterações, versão HTM 12.1.

Tabela 5.11: Comparação do método de predição inter-vistas proposto em relação ao 3D-HEVC considerando a codificação da Vista 2.

Sequência	QP	Δ Taxa de Bits [%]	Δ PSNR [dB]	Δ Tempo [%]	BD-rateY [%]	BD-PSNRY [dB]
<i>Balloons</i>	25	0,61	-0,072	-24,52	1,5	-0,06
	30	0,66	-0,079	-26,47		
	35	0,73	-0,085	-28,25		
	40	0,82	-0,098	-28,93		
<i>Newspaper</i>	25	0,64	-0,079	-25,97	1,8	-0,07
	30	0,77	-0,091	-26,03		
	35	0,94	-0,098	-26,80		
	40	1,06	-0,108	-28,92		
<i>Kendo</i>	25	0,56	-0,069	-24,57	1,6	-0,06
	30	0,67	-0,087	-26,92		
	35	0,75	-0,096	-27,42		
	40	0,81	-0,103	-30,32		
<i>GTFly</i>	25	0,68	-0,068	-25,16	2,0	-0,05
	30	0,72	-0,072	-27,22		
	35	0,79	-0,083	-28,23		
	40	0,88	-0,091	-31,15		
<i>PoznanStreet</i>	25	0,78	-0,054	-24,30	1,9	-0,05
	30	0,82	-0,065	-25,74		
	35	0,91	-0,078	-27,43		
	40	1,01	-0,093	-29,54		
<i>UndoDancer</i>	25	0,60	-0,051	-27,07	2,1	-0,04
	30	0,67	-0,058	-29,18		
	35	0,79	-0,074	-31,21		
	40	0,92	-0,081	-32,83		
Média		0,77	-0,081	-27,67	1,8	-0,06

De acordo com os resultados de simulação apresentados na Tabela 5.11, o método proposto nesta seção consegue uma redução no tempo de codificação de até 32,83% em relação à codificação usando a extensão 3D-HEVC (aproximadamente 28% em

média), apresentando uma degradação mínima em termos de eficiência de codificação (0,06 dB em BD-PSNR).

5.2.2. Redução de Complexidade da Codificação Intra dos Mapas de Profundidade

Assim como no HEVC, a codificação intra do 3D-HEVC também utiliza um processo de decisão de modo simplificado (RMD), o qual é composto pelas seguintes etapas. Inicialmente, os 35 modos intra convencionais do HEVC são testados para selecionar os modos que irão compor o subconjunto de modos de predição candidatos. Este subconjunto é formado pelos modos com menor valor de SATD [17], seguindo a ordem de processamento apresentada na Figura 5.10.

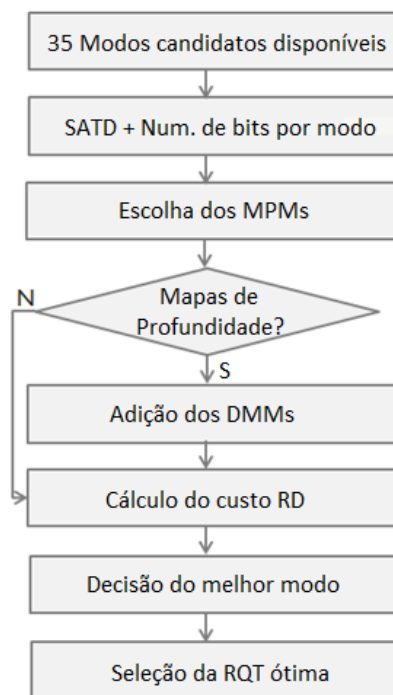


Figura 5.10: Processo de decisão de modo intra do 3D-HEVC.

De seguida é efetuada a escolha dos modos mais prováveis (MPMs), os quais e são adicionados ao subconjunto de modos a ser avaliado no cálculo do custo RD. O padrão HEVC define três MPMs. A escolha dos MPMs é baseada nos modos intra utilizados nos blocos vizinhos à esquerda e acima do bloco atual. Normalmente, os dois primeiros MPMs selecionados são os modos de ambos os blocos vizinhos a esquerda e superior e o terceiro MPM é definido como o modo Planar, modo DC ou modo angular 26, nesta ordem de

prioridade. No entanto, nos casos em que as PUs acima e à esquerda utilizam os mesmos modos intra, este modo e os dois modos direcionais mais próximos são escolhidos para compor o conjunto de MPMs [5].

Quando a trama, que está a ser comprimida é um mapa de profundidade, dois modos adicionais são considerados para serem incluídos no subconjunto de modos candidatos: os DMMs (*Wedgelet* e de Contorno).

Após a adição dos modos DMMs, o custo RD de cada modo de predição incluído no subconjunto de candidatos é calculado e o modo com o melhor desempenho (em termos de custo RD) é selecionado para codificar a PU. Finalmente, o resíduo obtido a partir do modo de predição intra selecionado é codificado usando uma *quadtrees* residual (RQT), concluindo a codificação intra do 3D-HEVC. Este processo é ilustrado na Figura 5.10.

Para facilitar a referência, a Tabela 5.12 lista os modos que são utilizados no processo de predição intra do 3D-HEVC juntamente com o número de índice correspondente.

Tabela 5.12: Modos de predição intra do 3D-HEVC.

Número do Modo	Nome do Modo
0	Planar
1	DC
2..34	Angular
35	DMM1 (<i>Wedgelet</i> - somente para prof.)
36	DMM4 (Contorno - somente para prof.)

Os DMMs foram incluídos na codificação dos mapas de profundidade como uma alternativa aos modos de predição intra convencionais adotados pelo HEVC visando aumentar a eficiência de codificação das PUs que possuem bordas bem definidas.

Esta seção apresenta um novo algoritmo para a decisão do modo intra durante a codificação dos mapas de profundidade do 3D-HEVC, o qual otimiza a escolha dos modos intra considerando a orientação de borda da informação de textura de cada PU de profundidade a ser codificada. Além disso, o algoritmo proposto explora a correlação entre os MPMs escolhidos e o modo intra selecionado como melhor modo para codificar a PU de profundidade, com o intuito de acelerar a decisão de modo e reduzir a complexidade computacional do processo de codificação intra de profundidade.

Este algoritmo explora duas características dos mapas de profundidade: a existência de grandes superfícies homogêneas e a ocorrência de bordas bem definidas,

conforme exposto na Seção 3.2.4.1. Em um primeiro momento, uma análise estatística foi realizada para determinar a percentagem de ocorrências dos DMMs que são escolhidos como o melhor modo intra quando o caso *all-intra* é considerado. Além disso, uma vez que os mapas de profundidade contêm grandes áreas homogêneas e lisas, é normal que o modo Planar seja selecionado como melhor modo muitas vezes. Por esta razão, a percentagem de ocorrências de modo Planar foi também registrada.

Esta análise estatística foi realizada utilizando as sequências de teste: ‘PoznanHall2’, ‘Shark’, ‘PoznanCarPark’ e ‘PoznanHall1’ com resolução de 1920x1088 pixels, especificadas em [86] e [102]. As simulações foram realizadas utilizando a configuração *all-intra* e os seguintes QPs: 25, 30, 35 e 40. Os resultados desta análise, considerando-se os resultados médios para todas as sequências de vídeo testadas, são apresentados na Tabela 5.13.

Tabela 5.13: Porcentagem de ocorrências dos modos intra no 3D-HEVC.

QP	DMMs [%]	Modo Planar [%]	Outros modos [%]
25	2,17	50,84	46,99
30	1,98	72,60	25,42
35	0,94	87,62	11,45
40	0,37	93,80	5,84
Méd.	1,37	76,22	22,43

A partir desses resultados, podemos concluir que, em média, os DMMs são escolhidos em apenas 1,37% dos casos. Por outro lado, conforme esperado, o modo Planar é selecionado na maioria dos casos, tendo em vista o grande número de regiões homogêneas presentes nos mapas de profundidade. Logo após esta análise, outro teste foi realizado com a intenção de avaliar a correlação entre a escolha do modo Planar como um dos MPMs e sua escolha como o melhor modo intra. Os resultados desta experiência são apresentados na Tabela 5.14.

Tabela 5.14: Correlação entre os MPMs e a escolha do modo Planar como melhor modo intra.

QP	Modo Planar = MPM [%]	Modo Planar ≠ MPM [%]
25	64,51	35,49
30	73,69	26,31
35	87,16	12,84
40	93,04	6,96
Méd.	79,6	20,4

A partir destes resultados, concluiu-se que, na maioria dos casos, quando o modo Planar é um dos MPMs selecionados, ele também é escolhido como o melhor modo intra. Além disso, a direção intra está fortemente correlacionada com a direção de borda nos dados dos mapas de profundidade [61]. Estas observações levaram ao desenvolvimento do algoritmo proposto, o qual explora as características da textura dos blocos de profundidade (bordas bem definidas e áreas homogêneas) e a relação entre os MPMs e o melhor modo intra selecionado, visando acelerar a decisão de modo da codificação intra de profundidade.

A Figura 5.11 apresenta o fluxograma do algoritmo proposto, o qual será explicado passo a passo nos parágrafos seguintes desta seção.

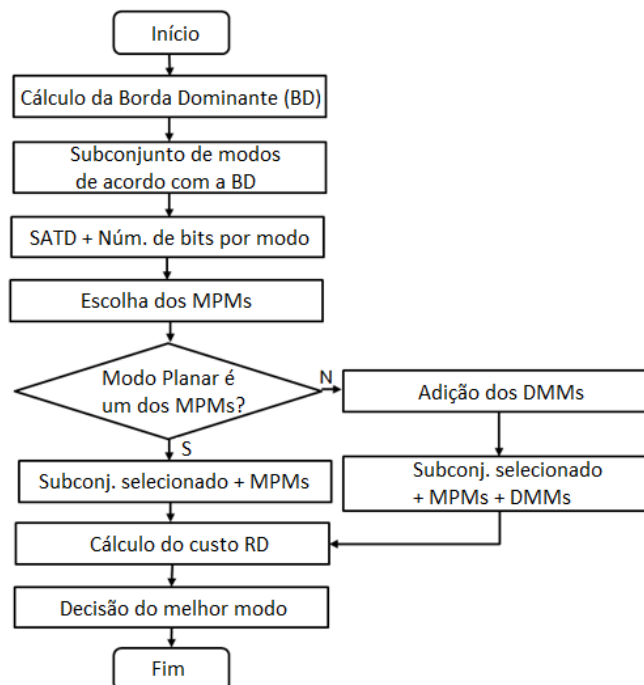


Figura 5.11: Fluxograma do algoritmo rápido para a decisão de modo intra dos mapas de profundidade.

No primeiro passo, a textura das PUs de profundidade é analisada para determinar a orientação de borda dominante de cada PU. Tal orientação é então utilizada para selecionar o subconjunto de modos que serão avaliados no procedimento de codificação intra dos mapas de profundidade. Cinco indicadores de orientação de borda são avaliados: horizontal, vertical, diagonal de 45° , diagonal de 135° e não-direcional [85]. Conforme explicado na Seção 4.1.1, a orientação de borda dominante de cada bloco 4×4 é calculada com base nos valores de luminância dos seus pixels, através da divisão do bloco 4×4 de amostras de luminância em quatro blocos de tamanho 2×2 , conforme ilustrado na

Figura 5.12. Em seguida, os cinco indicadores de força de orientação de borda são calculados, conforme explicado em detalhe na Seção 4.1.1.

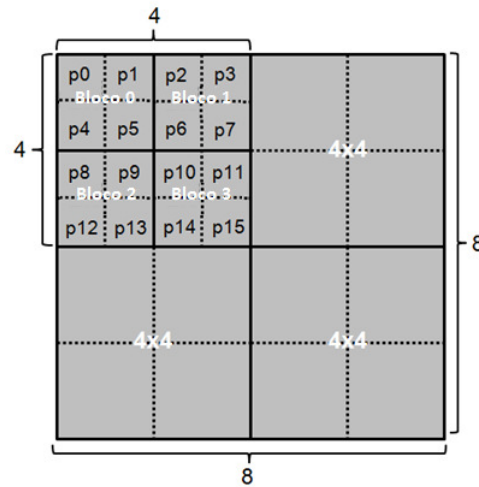


Figura 5.12: Cálculo de borda de uma PU 8x8.

A orientação de borda dominante é determinada a partir das forças de orientação de borda calculadas, de modo que a orientação com maior valor (ou força) é escolhida como dominante, de acordo com a equação (4.7) apresentada na Seção 4.1.1.

No segundo passo do fluxograma da Figura 5.11, o subconjunto de modos a ser avaliado na decisão do modo intra é definido considerando a força de borda.

Cinco subconjuntos de modos angulares são definidos, cada um composto por nove modos angulares selecionados a partir do conjunto completo de 33 modos angulares adotado pelo HEVC, conforme explicado na Seção 4.1.1. Cada subconjunto de nove modos corresponde a uma das cinco orientações de borda definidas anteriormente (horizontal, vertical, 45°, 135° e não-direcional), conforme apresentado na Figura 4.15, a qual é rerepresentada de forma semelhante na Figura 5.13, para facilitar a leitura da tese. O processo de predição intra da profundidade utiliza apenas um dos cinco subconjuntos de modos de acordo com sua orientação da borda dominante.

Portanto, utilizando a textura das PUs dos mapas de profundidade para selecionar um subconjunto de nove modos angulares de acordo com a orientação de borda dominante, já não é necessário avaliar os 33 modos angulares no primeiro passo de decisão do modo intra de profundidade, mas apenas os nove modos que fazem parte do subconjunto selecionado. No terceiro passo do algoritmo, o cálculo de SATD é feito, e no quarto passo, a escolha dos MPMs é realizada, conforme ilustrado no fluxograma da Figura 5.11.

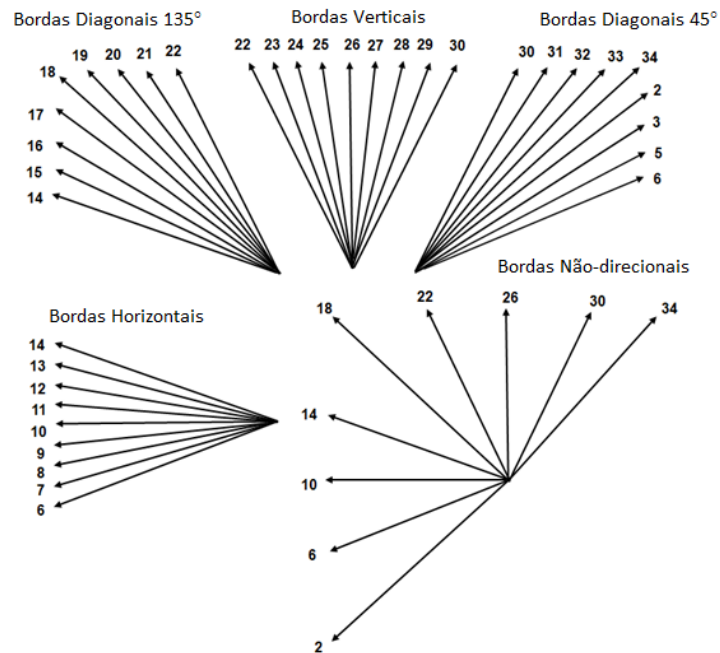


Figura 5.13: Subconjuntos de modos para cada uma das cinco orientações de borda.

No quinto passo, considerando os resultados da análise estatística apresentada na Tabela 5.14, um teste é feito para verificar se algum dos MPMs selecionados é o modo Planar. Se esse teste for verdadeiro, os DMMs não são adicionados ao subconjunto de modos candidatos a serem avaliados no cálculo do custo RD. No entanto, se o modo Planar não estiver entre os MPMs escolhidos, os DMMs são adicionados ao subconjunto de candidatos, conforme apresentado na Figura 5.11.

No sexto passo do algoritmo, o subconjunto de modos está definido e os MPMs e DMMs (de acordo com o resultado do teste do quinto passo) são adicionados a este subconjunto. Finalmente, os próximos passos do algoritmo são o cálculo do custo RD e a decisão do melhor modo intra para a PU do mapa de profundidade.

Para avaliar a redução da complexidade computacional do algoritmo proposto e o seu efeito sobre a performance de codificação, um grupo de experimentos foi realizado utilizando o HTM12.1 (software de referência do 3D-HEVC), o qual foi modificado para incluir o método descrito neste seção.

Os experimentos foram conduzidos usando as sequências de teste apresentadas na Tabela 4.1. As simulações foram realizadas usando a configuração *all-intra*, os valores de QPs aplicados às vistas da textura foram 25, 30, 35 e 40 e, os QPs utilizados para codificar os mapas de profundidade foram 34, 39, 42 e 45 conforme recomendado em [86].

Tabela 5.15 apresenta os resultados da codificação, comparando o desempenho e a complexidade do codificador 3D-HEVC utilizando o algoritmo proposto, em relação ao 3D-HEVC sem modificações. Nesta tabela, a coluna "Apenas Vídeo" apresenta os resultados de *BD-Rate* que correspondem à compressão das vistas de textura, e a coluna "Apenas Síntese" apresenta os resultados de *BD-Rate* das vistas sintetizadas. Os valores para as vistas sintetizadas são apresentados pois o método proposto é aplicado na codificação dos mapas de profundidade, os quais tem influência direta na qualidade das vistas sintetizadas, e não são visualizados de forma independente no vídeo codificado. A síntese das vistas foi efetuada utilizando o algoritmo VSRS (*View Synthesis Reference Software*) [11], descrito em detalhe em [103]. O cálculo de *BD-rate* das vistas sintetizadas considera os valores de PSNR das vistas sintetizadas, e a soma das taxas de bits dos mapas de profundidade e vistas de textura, conforme apresentado na planilha Excel disponibilizada pela JCT-3V em [86].

A coluna "Tempo de Codificação" da Tabela 5.15 corresponde ao tempo de processamento do codificador 3D-HEVC, e a coluna "Redução no Tempo Total" apresenta a redução de complexidade alcançada pelo algoritmo proposto, comparando o tempo de processamento do codificador 3D-HEVC original em relação ao codificador 3D-HEVC incluindo a solução proposta. Por fim, a coluna "Redução no Tempo da Profund." apresenta a redução de complexidade considerando apenas a codificação dos mapas de profundidade. Tal coluna foi adicionada a essa tabela, tendo em vista que o algoritmo proposto nessa seção tem o objetivo específico de reduzir o tempo de codificação intra dos mapas de profundidade.

Tabela 5.15: Resultados de BD-rate e tempo de processamento para 3 vistas considerando o método proposto para a codificação intra dos mapas de profundidade, caso *all-Intra*.

Sequência	Apenas Vídeo BD-Rate [%]	Apenas Síntese BD-Rate [%]	Tempo de Codificação [%]	Redução no Tempo Total [%]	Redução no Tempo da Profund. [%]
Balloons	0,21%	1,06%	86,5%	13,5%	19,3%
Kendo	0,27%	0,60%	86,5%	13,5%	20,3%
Newspaper	0,17%	0,71%	88,3%	11,7%	16,9%
GTFly	0,05%	0,02%	95,7%	4,3%	8,4%
PoznanStreet	0,11%	0,07%	93,4%	6,6%	11,4%
UndoDancer	0,08%	0,67%	94,7%	5,3%	10,7%
1024x768	0,22%	0,79%	87,1%	12,9%	18,8%
1920x1088	0,11%	0,25%	94,6%	5,4%	10,2%
Média	0,15%	0,52%	90,8%	9,2%	15,0%

De acordo com os resultados experimentais mostrados na Tabela 5.15, considerando a codificação 3D-HEVC completa (vistas de textura e mapas de profundidade) o algoritmo proposto permite alcançar uma redução de complexidade de 9,2% em média, e considerando somente a codificação dos mapas de profundidade, o algoritmo proposto alcança 15% de redução de complexidade, em média. Com relação à eficiência de codificação, foi observado um aumento médio em BD-rate de 0,15% para a codificação das vistas de textura do vídeo, e de 0,52% para a codificação das vistas sintetizadas.

5.2.3. Redução de Complexidade da Codificação Intra de Textura e Profundidade

Esta seção apresenta um algoritmo rápido para a seleção de modo da codificação intra do 3D-HEVC. O algoritmo proposto realiza uma análise da textura das PUs nas vistas de textura e mapas de profundidade usando um filtro de Sobel [104] para detectar as bordas e determinar suas localizações. A partir desta informação de borda, as direções angulares são calculadas com base no gradiente dos pixels da PU. As direções angulares com maior frequência de ocorrência são, então, identificadas e avaliadas na etapa de otimização da taxa de distorção (RDO). Uma vez que estas direções angulares são menos numerosas do que o conjunto completo de modos suportados pelo 3D-HEVC, uma economia computacional significativa é obtida com uma perda insignificante em eficiência de codificação, conforme será apresentado a seguir.

Tendo em vista o processo de decisão do modo intra do 3D-HEVC explicado em detalhe na seção anterior (Seção 5.2.2) e apresentado na Figura 5.10, o algoritmo proposto nesta seção visa reduzir o tempo consumido por esse processo através da otimização do subconjunto de modos candidatos usando técnicas de detecção de borda da textura da PU. Tal algoritmo, tem o objetivo de contribuir significativamente para a redução da complexidade global do codificador 3D-HEVC e, conseqüentemente, tornar a sua utilização mais apropriada para dispositivos com menor disponibilidade de energia e/ou poder computacional.

As áreas homogêneas e bordas bem definidas de uma imagem podem ser identificadas por meio de análise da textura. Em particular, a informação de gradiente de cada componente de textura da PU pode ser utilizado para acelerar a seleção do modo

intra. No entanto, para ser útil na redução da complexidade do procedimento de codificação intra esta análise da textura não deve introduzir uma grande sobrecarga computacional, nem diminuir significativamente o desempenho de codificação do vídeo.

O algoritmo apresentado neste trabalho utiliza um método de análise da textura eficiente, o qual é baseado na extração de bordas para identificar a direção de borda dominante da PU. Esta informação de direcionalidade é então utilizada para determinar um pequeno número de modos candidatos promissores, resultando em uma redução da complexidade computacional do RMD.

5.2.3.1. Determinação de Borda

Para implementar a técnica de seleção rápida do modo intra proposta, inicialmente um filtro de Sobel é aplicado aos pixels da PU que está a ser codificada. O filtro Sobel foi escolhido por possuir uma baixa complexidade computacional e um bom desempenho [105]. Outros filtros de detecção de borda, tais como LOG (*Laplacian of Gaussian*) e filtro de *Canny* também poderiam ser usados, mas com um custo em complexidade computacional mais elevado [106].

O filtro de Sobel realiza um cálculo de gradiente espacial em 2D sobre todas as posições de pixel de uma imagem e produz um mapa de intensidade onde valores elevados indicam regiões de alta frequência espacial, as quais usualmente indicam a presença de bordas [107].

Assim, nesta seção, a fim de extrair a informação de borda da PU a ser predita, um mapa de bordas da textura da PU é calculado aplicando-se os operadores de Sobel. Cada pixel da PU está associado a um gradiente, identificado pela sua amplitude e direção.

Do ponto de vista de processamento da imagem, o operador de Sobel consiste de duas máscaras de convolução de 3x3 pixels conforme mostrado na Figura 5.14, as quais calculam o grau de variação nas direções horizontal (G_x) e vertical (G_y).

-1	0	1		-1	-2	-1
-2	0	2		0	0	0
-1	0	1		1	2	1
G_x			G_y			

Figura 5.14: Máscaras do operador de Sobel.

As máscaras de convolução (horizontal e vertical) são aplicadas à imagem separadamente, conforme ilustrado na Figura 5.15, de modo que para cada pixel $p_{i,j}$, um vetor de borda correspondente é calculado como segue:

$$\vec{G}_{i,j} = \{Gx_{i,j}, Gy_{i,j}\}, \text{ onde:} \quad (5.1)$$

$$Gx_{i,j} = p_{i+1,j-1} + 2p_{i+1,j} + p_{i+1,j+1} - p_{i-1,j-1} - 2p_{i-1,j} - p_{i-1,j+1} \quad (5.2)$$

$$Gy_{i,j} = p_{i-1,j-1} + 2p_{i,j-1} + p_{i+1,j-1} - p_{i-1,j+1} - 2p_{i,j+1} - p_{i+1,j+1} \quad (5.3)$$

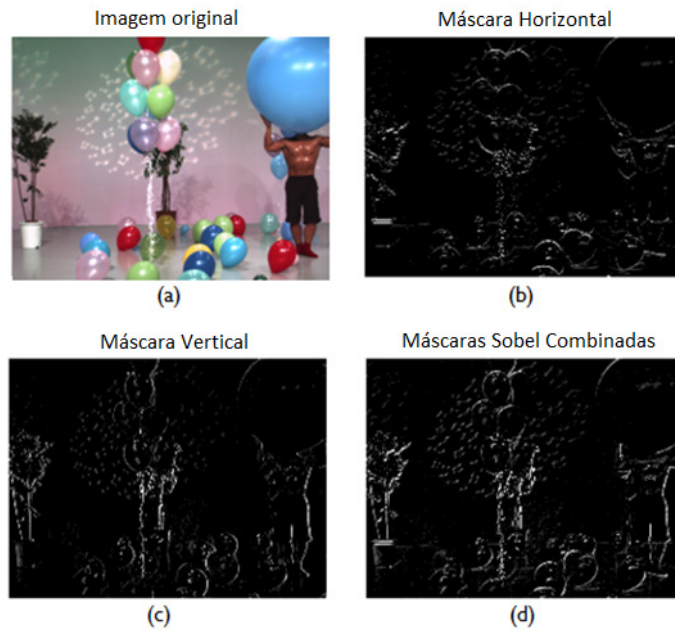


Figura 5.15: Exemplo de aplicação das máscaras Sobel utilizando uma trama do vídeo Balloons: Imagem original (a), Máscara de Sobel Horizontal (b), Máscara de Sobel Vertical (c), Máscaras de Sobel Combinadas (d).

A amplitude e a orientação do vetor de borda são então calculados a partir dos componentes horizontais e verticais obtidos usando o filtro de Sobel, de acordo com as equações (5.4) e (5.5):

$$Amp |\vec{G}_{i,j}| = \sqrt{(Gx_{i,j})^2 + (Gy_{i,j})^2} \quad (5.4)$$

$$GradDir (\vec{G}_{i,j}) = \arctan\left(\frac{Gy_{i,j}}{Gx_{i,j}}\right) \quad (5.5)$$

Contudo, uma amplitude aproximada, definida através da equação (5.6) é normalmente utilizada, uma vez que é menos dispendiosa computacionalmente do que a expressão (5.4) e os seus resultados são muito semelhantes.

$$Amp |\vec{G}_{i,j}| = |Gx_{i,j}| + |Gy_{i,j}| \quad (5.6)$$

Além disso, uma simplificação do cálculo de $GradDir$ é também usada, aproximando-se $GradDir$ por $G_{y_{i,j}}/G_{x_{i,j}}$, com o intuito de otimizar o algoritmo para compor um histograma de direções de borda, conforme explicado a seguir.

5.2.3.2. Seleção dos Modos Candidatos

A fim de selecionar os modos intra mais promissores para compor o subconjunto de modos candidatos no processo de RMD, um histograma de direções de borda é calculado a partir dos pixels da PU através da soma das amplitudes dos pixels com mesma direção de borda na PU. Assim, a direção de borda dominante de cada PU é calculada. Uma descrição detalhada deste processo será explicada a seguir.

Na Figura 5.16 representa-se de forma gráfica a relação entre cada predição de modo intra angular e sua direção correspondente ϕ .

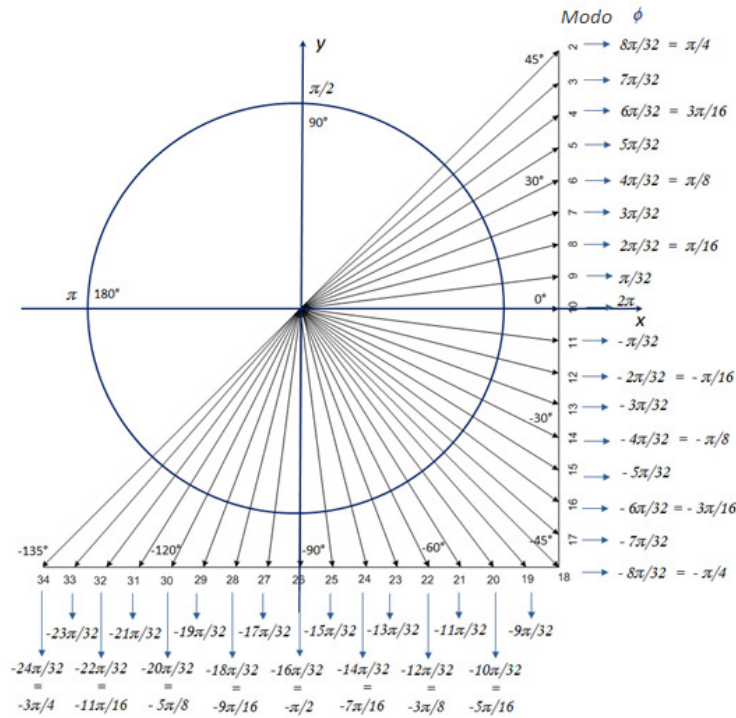


Figura 5.16: Modos angulares e direção ϕ associada.

Para cada pixel da PU, inicialmente a direção de gradiente $GradDir$ do vetor de borda é calculada. Logo após, uma direção angular ψ é calculada como a mais próxima de $GradDir$, a partir da diferença entre a direção ϕ associada aos modos intra angulares e o $GradDir$, tal como apresentado em (5.7):

$$\psi = \arg \min_{\phi \in [-3\pi/4, \pi/4]} |\phi - GradDir| \quad (5.7)$$

A direção do gradiente calculado numa posição de pixel corresponde à direção de variação máxima e, a direção perpendicular corresponde à direção de variação mínima [108] (conforme ilustrado na Figura 5.17). Assim, o ψ calculado corresponde à direção de variação máxima na posição do pixel, e o modo intra angular com direção perpendicular em relação à ψ corresponderá à variação mínima.

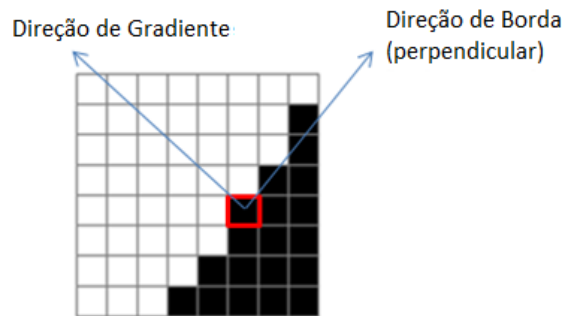


Figura 5.17: Direção de gradiente e direção de borda.

Para construir o histograma de direção de modo, a magnitude dos gradientes dos pixels com a mesma direção de modo intra ϕ da textura da PU são adicionados, de acordo com a equação *AmpSum* apresentada em (5.8). Em (5.8) *ImPU* corresponde ao subconjunto de pixels da PU a ser codificada, que possuem a mesma direção de modo intra, perpendicular à ψ .

$$\text{AmpSum}(\phi) = \sum_{i,j \in \text{ImPU}(\phi)} \text{Amp} |\vec{G}_{i,j}| \quad (5.8)$$

Ao calcular esta soma levando em consideração todos os pixels da PU a ser codificada, temos um histograma de direções de modo a partir do qual podemos escolher os modos angulares candidatos mais prováveis como aqueles que correspondem aos picos do histograma. Um exemplo de um histograma de direções de modo para uma PU de tamanho 32x32 é apresentado na Figura 5.18. No caso do exemplo, podemos concluir que a PU apresenta bordas que se destacam na direção vertical, uma vez que os modos 25, 26 e 27 têm valores de *AmpSum* significativos.

O histograma de direções é utilizado para acelerar a seleção do modo intra, através da escolha de um subconjunto de modos candidatos reduzido, selecionado de acordo com o *AmpSum*. No método proposto o subconjunto é constituído pelos nove modos com maiores valores de *AmpSum*. A decisão de limitar o conjunto a nove modos baseou-se nas explorações apresentadas na Seção 4.1.1, onde se verificou que o conjunto com nove modos a serem avaliados na predição intra apresentou o melhor compromisso

entre eficiência de codificação e complexidade computacional. No caso do exemplo apresentado na Figura 5.18, os nove modos direcionais selecionados para serem avaliados no processo de RMD seriam os modos 21, 22, 23, 25, 26, 27, 28, 30 e 34. Os modos Planar e DC são sempre incluídos no subconjunto de candidatos e a adição de DMMs é realizada conforme definido no software de referência do 3D-HEVC [91]. Assim, usando este conjunto de tamanho reduzido já não é necessário avaliar todos os 33 modos angulares originais durante a pesquisa RDO para escolher o melhor modo de predição, reduzindo-se significativamente a complexidade deste processo de pesquisa.

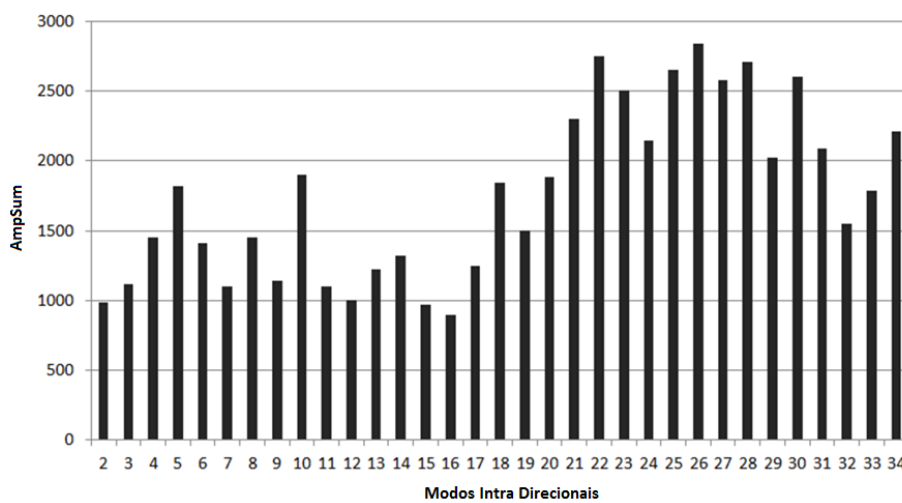


Figura 5.18: Exemplo de um histograma de direções de modo para uma PU 32x32.

Um grupo de experimentos foram realizados para avaliar a redução de complexidade computacional conseguida usando o algoritmo proposto e o seu efeito sobre o desempenho da codificação. Estes experimentos foram realizados usando o HTM12.1 [91], modificado para incluir o método proposto nesta seção. Os experimentos foram conduzidos utilizando as sequências de teste apresentadas na Tabela 4.1, as quais são rerepresentadas na Tabela 5.16 para facilitar a leitura e a visualização dos índices das vistas que poderão ser referenciados durante a apresentação dos resultados.

Cada sequência de teste contém três vistas, cada uma consistindo de informação de textura e mapa de profundidade correspondente. Na Tabela 5.16, os números apresentados na coluna "3 Vistas de Entrada" indicam os índices das vistas/câmeras, e a ordem de codificação dessa vistas é central-esquerda-direita. As simulações foram realizadas usando os QPs 25, 30, 35 e 40 aplicados às vistas da textura e, 34, 39, 42 e 45 aplicados aos mapas de profundidade [86].

Tabela 5.16: Características das sequências de teste.

Sequência	Taxa de Tramas (fps)	Número de Tramas	3 Vistas de Entrada	Resolução (pixels)
Balloons	30	100	1-3-5	1024x768
Newspaper	30	100	1-3-5	1024x768
Kendo	30	100	2-4-6	1024x768
GTFly	25	100	9-5-1	1920x1080
PoznanStreet	25	100	5-4-3	1920x1080
UndoDancer	25	100	1-5-9	1920x1080

Os resultados foram obtidos como segue. Inicialmente, um grupo de testes foi realizado, usando as sequências indicadas na Tabela 5.16. Tais sequências foram codificadas usando o software de referência HTM12.1 sem alterações considerando o caso *all-intra* e, a seguir, as mesmas sequências foram codificadas usando o algoritmo proposto implementado no HTM12.1.

A Tabela 5.17 apresenta os resultados de codificação detalhados, comparando o desempenho e a complexidade do algoritmo proposto com os resultados obtidos utilizando o codificador 3D-HEVC sem alterações.

Tabela 5.17: Resultados de BD-rate e tempo de processamento para 3 vistas considerando o método proposto para a codificação intra das vistas de textura e profundidade, caso *all-Intra*.

Sequência	Apenas Vídeo BD-Rate [%]	Apenas Síntese BD-Rate [%]	Tempo de Codificação [%]	Redução no Tempo Total [%]
Balloons	0,34%	0,52%	74,1%	25,9%
Kendo	0,38%	0,62%	73,7%	26,3%
Newspaper	0,31%	0,44%	75,3%	24,7%
GTFly	0,39%	0,62%	72,2%	27,8%
PoznanStreet	0,23%	0,29%	76,0%	24,0%
UndoDancer	0,27%	0,18%	72,7%	27,3%
1024x768	0,34%	0,53%	74,4%	25,6%
1920x1088	0,30%	0,36%	73,6%	26,4%
Média	0,32%	0,45%	74%	26%

De acordo com os resultados experimentais mostrados na Tabela 5.17, o algoritmo proposto permite atingir uma redução de complexidade média de 26%, considerando o tempo total de codificação (vistas de textura e mapas de profundidade).

Com relação à eficiência de codificação, os resultados foram avaliados através do método Bjontegaard delta [40][41]. Um aumento no *BD-rate* de 0,32%, em média, foi observado para o vídeo somente (vistas de textura), e um aumento de 0,45%, em média, para as vistas sintetizadas. Como a escolha dos modos candidatos é parte do processo RMD e este processo é aplicado a ambos os componentes de textura e profundidade de cada vista, os resultados em termos de desempenho e complexidade refletem essa otimização do RMD.

Os resultados obtidos dependem das características do vídeo multivista, tais como a distância entre as câmeras usadas para capturar as vistas do vídeo, ambiente de gravação interno ou externo, o movimento dos objetos na cena, e utilização de computação gráfica para produzir a sequência (por exemplo: UndoDancer e GTFly).

As Figuras 5.19 e 5.20 apresentam os resultados de desempenho e complexidade obtidos através da codificação da sequência de teste GTFly (1920x1088 pixels).

Na Figura 5.19, as curvas RD para a vista de textura com número de índice 5 da sequência GTFly são apresentadas para ilustrar os resultados obtidos. Cada ponto nas curvas representa um valor de QP. A partir desta figura é possível concluir que a eficiência de codificação do algoritmo proposto nesta tese é muito semelhante à obtida com o HTM12.1, introduzindo uma perda mínima em desempenho. Esta perda é quase imperceptível na Figura 5.19. Os resultados completos correspondentes à sequência de vídeo GTFly são mostrados na Tabela 5.17.

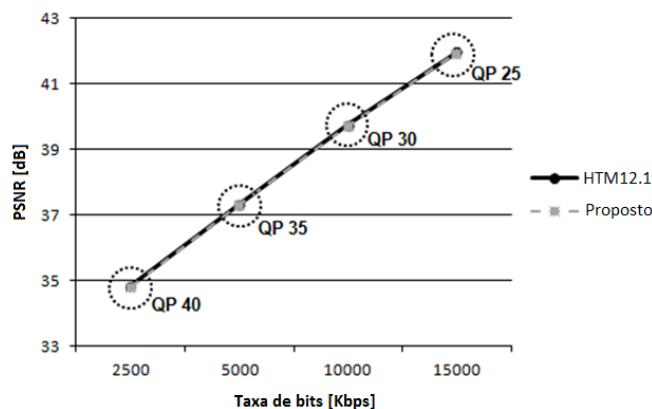


Figura 5.19: Curvas RD do algoritmo proposto em relação ao HTM 12.1 (GTFly 1920x1080 pixels).

A Figura 5.20 mostra os tempos de codificação versus a taxa de bits para os diferentes valores de QP comparando o algoritmo apresentado nesta seção e os resultados

do HTM12.1. Os resultados no gráfico mostram que o algoritmo proposto consegue uma economia significativa de tempo em relação ao HTM12.1.

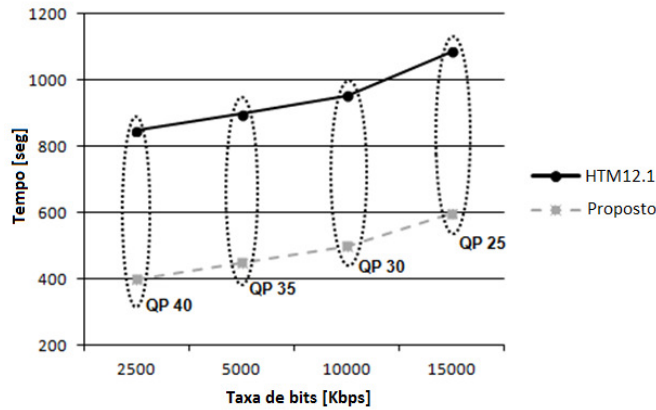


Figura 5.20: Comparação do tempo de codificação do algoritmo proposto em relação ao HTM12.1 (GTFLy 1920x1080 pixels).

As Figuras 5.21 e 5.22 apresentam os resultados correspondentes à eficiência de codificação e a complexidade computacional para a sequência Newspaper (1024x768 pixels).

Na Figura 5.21 as curvas RD para a vista de textura com número de índice 4 da sequência Newspaper são apresentadas. A partir desta figura, é possível notar que o algoritmo proposto apresenta perdas de desempenho RD insignificantes.

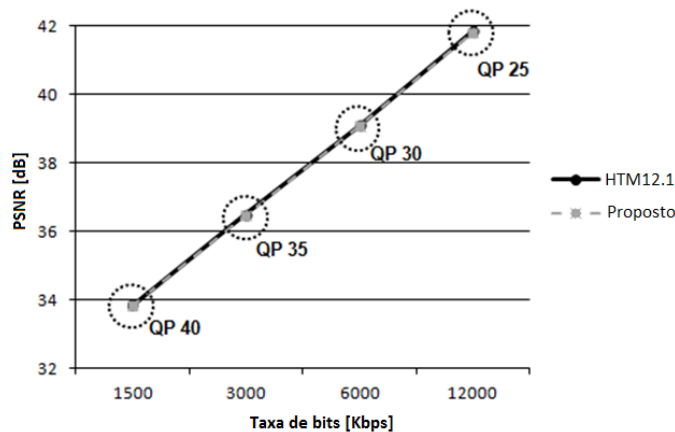


Figura 5.21: Curvas RD do algoritmo proposto em relação ao HTM 12.1 (Newspaper 1024x768 pixels).

Na Figura 5.22 os tempos de codificação para a mesma sequência Newspaper são mostrados, e pode-se observar que o algoritmo proposto é bastante menos complexo, computacionalmente, que o codificador 3D-HEVC original.

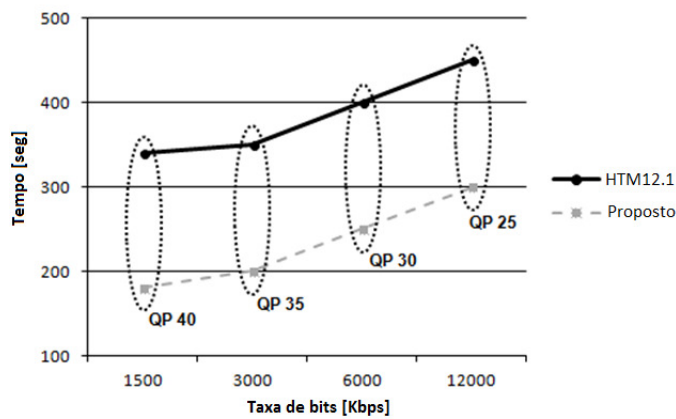


Figura 5.22: Comparação do tempo de codificação do algoritmo proposto em relação ao HTM12.1 (Newspaper 1024x768 pixels).

5.3. Análise de Resultados e Comparações

Neste capítulo, inicialmente, foi apresentado um método para predição inter-vista do modo intra, o qual explora a correlação entre a textura de vistas adjacentes, acelerando o processo de codificação intra de vistas vizinhas. Nesse método, as vistas base (vistas pares) foram codificadas usando o algoritmo rápido de decisão do modo de predição intra proposto na Seção 5.1.1, e a vista dependente (vista ímpar) foi codificada utilizando os modos intra selecionados nas vistas pares como modos candidatos no seu processo de predição intra. Assim, para codificar as vistas dependentes, o procedimento de codificação intra convencional, no qual todos os modos direcionais são testados para a escolha daquele com melhor desempenho já não é necessário, mas apenas o método de predição inter-vista proposto.

De acordo com os resultados das simulações, o método de predição inter-vista do modo intra alcançou uma redução média no tempo de codificação de 45,44% em relação ao HEVC *Simulcast*, com uma degradação mínima em eficiência de codificação de 0,08dB em BD-PSNR, em média.

O segundo método proposto neste capítulo é um algoritmo de predição inter-vistas da estrutura da árvore de codificação, o qual utiliza as profundidades máximas das árvores de CUs definidas durante a codificação da vista base, como um limite a ser usado na codificação das vistas dependentes. Assim, as comparações do custo de taxa de distorção (RD) de todas as estruturas de profundidade da árvore de CU deixam de ser necessárias para codificar as vistas dependentes, reduzindo assim a complexidade da codificação. Os resultados experimentais desse método demonstraram que quando

comparado com o *HEVC simulcast*, o tempo de processamento da codificação foi reduzido em cerca de 29% com uma perda negligenciável em eficiência de codificação.

O terceiro método, um algoritmo rápido para a decisão de profundidade da árvore de CUs, que analisa as profundidades máximas da árvore de CUs definidas durante a codificação das duas primeiras vistas a fim de acelerar a codificação da terceira é apresentado e analisado. Os resultados experimentais mostraram que quando comparado ao 3D-HEVC, este método atinge uma redução no tempo de codificação de até 32,83% (28% em média), com uma degradação mínima em termos de eficiência de codificação, cerca de 0,06 dB em BD-PSNR.

A seguir, um algoritmo rápido para a decisão do modo intra visando reduzir a complexidade de codificação dos mapas de profundidade do 3D-HEVC, foi apresentado. O algoritmo proposto explorou duas características importantes dos mapas de profundidade: a existência de grandes áreas homogêneas, as quais foram exploradas através da correlação entre o modo Planar e a escolha dos MPMs; e a ocorrência de bordas bem definidas nos mapas de profundidade, as quais são identificadas através da análise da textura dos blocos de profundidade, determinando a orientação de borda dominante para reduzir o número de modos a serem avaliados na decisão de modo intra da profundidade. Os resultados experimentais relativos a esse algoritmo, mostraram que quando comparado ao 3D-HEVC não modificado, o tempo médio de processamento da codificação dos mapas de profundidade foi reduzido em 15%, com uma perda mínima em eficiência de codificação de 0,52% em BD-rate para as vistas sintetizadas.

Finalmente, um método rápido para seleção do modo intra foi desenvolvido, com o objetivo de reduzir a complexidade da codificação intra do 3D-HEVC, considerando as vistas de textura e mapas de profundidade. Esse método tem por base uma análise da textura com um filtro Sobel, cujos resultados permitem determinar as orientações mais representativas da textura, informação essa que de seguida é usada para construir um subconjunto reduzido de modos candidatos a serem avaliados no cálculo de otimização da taxa de distorção (RDO).

Os resultados experimentais mostraram que quando comparada com o 3D-HEVC, a complexidade computacional da codificação intra foi reduzida em 26,2%, em média, considerando de vistas textura e mapas de profundidade, com uma perda mínima em eficiência de codificação de 0,30% em BD-rate, em média.

Alguns algoritmos têm sido publicados com objetivo de simplificar computacionalmente a codificação 3D-HEVC. Em [109], Mora et al. apresentam um método de predição inter-componente que explora as semelhanças entre os componentes de textura e profundidade.

Nesse método, se a profundidade é codificada antes da textura, a *quadtree* de textura é inicializada a partir da *quadtree* de profundidade já codificada. Por outro lado, se a textura é codificada primeiro, a profundidade da *quadtree* é limitada à *quadtree* de textura codificada, forçando uma CU de profundidade a ser menos ou igualmente dividida de acordo com sua CU de textura co-localizada. Em [110][111], um novo método de simplificação da codificação dos mapas de profundidade é apresentado, o qual explora as correlações entre os modos intra de textura e profundidade com o objetivo de aperfeiçoar a eficiência de codificação das vistas sintetizadas.

No entanto, apenas alguns algoritmos têm sido propostos para reduzir a complexidade de codificação intra [111] - [114]. O algoritmo proposto em [111] escolhe os DMMs para serem usados de acordo com o valor do primeiro modo na lista de pesquisa completa RD (*Full-RD*) e considerando a variância do bloco de profundidade a ser codificado. Em [112], uma técnica que classifica a PU para ser codificada como uma borda ou uma região constante é proposto, de modo que, se a PU é classificada como uma região homogênea, os DMMs não são incluídos no processo de predição intra de profundidade.

O algoritmo proposto por [113] investiga as correlações entre os coeficientes de Hadamard das PUs nos níveis de profundidade adjacentes, e com base nessas correlações ele realiza uma classificação de borda da PU, excluindo seletivamente os DMMs da decisão do modo intra de profundidade. Em [114], um algoritmo rápido de escolha dos DMMs é proposto para acelerar o processo de codificação intra da profundidade, o qual descarta o cálculo de custo RD dos DMMs de acordo com os MPMs selecionados. O algoritmo proposto em [31] é o método descrito na Seção 5.2.2, o qual apresenta uma decisão rápida do modo intra para a codificação dos mapas de profundidade. Tal método otimiza a escolha dos modos considerando a direção de borda das PUs de profundidade e, explora a correlação entre os MPMs e o modo Planar para decidir se os DMMs serão ou não adicionados ao processo de codificação intra.

Comparando-se o algoritmo proposto na Seção 5.2.3 para redução da complexidade da codificação intra das vistas de textura e mapas de profundidade com os

trabalhos relacionados, é possível concluir que o nosso trabalho apresenta uma maior economia de tempo de processamento do que a solução apresentada em [111]. Uma vez que o método desenvolvido em [111] foi adotado na versão 9.1 do HTM (e incorporado às versões seguintes) e, o algoritmo proposto no presente trabalho foi testado usando a versão 12.1 do HTM, nosso algoritmo adiciona economia de tempo àquela introduzida por [111].

No que se refere ao método proposto por [112], tendo em vista que o mesmo não apresenta resultados considerando a configuração *all-intra*, não é possível comparar a redução de complexidade no tempo de codificação com o presente trabalho.

Quando comparado com [113], o algoritmo proposto na Seção 5.2.3 alcança uma redução de complexidade maior do que aquela reportada em [113], a qual corresponde a 22,19%, em média. No que diz respeito a eficiência de codificação, o algoritmo proposto em [113] apresenta um aumento na taxa de bits de 0,13% para as vistas sintetizadas quando comparado com HTM 9.1.

O algoritmo apresentado neste trabalho obtém resultados semelhantes aos apresentados em [114], o qual permite alcançar uma redução média de 27,8% no tempo de codificação, com um aumento na taxa de bits de 0,49% considerando as vistas sintetizadas.

Quando comparado com o método proposto na Seção 5.2.2, publicado em [31], o algoritmo proposto na Seção 5.2.3 ultrapassa a redução no tempo total de codificação de 9,2% apresentada em [31], com um aumento mínimo de 0,27% em BD-rate considerando as vistas sintetizadas.

6. CONCLUSÕES E TRABALHOS FUTUROS

Esta tese apresentou o desenvolvimento de soluções algorítmicas visando reduzir a complexidade da codificação intra nos cenários de vista única, utilizando o padrão HEVC e, no cenário multivista utilizando a extensão 3D-HEVC, a qual se encontra em fase final de padronização. Tais soluções alcançaram reduções significativas em complexidade com perdas insignificantes de eficiência de codificação.

Os algoritmos rápidos apresentados no Capítulo 4 e Seção 5.1.1 do Capítulo 5, concretizam a primeira contribuição citada na Seção 1.3.1, a qual visava reduzir a complexidade da predição intra-tramas do HEVC e de sua extensão multivista para vistas de textura. Os algoritmos desenvolvidos no Capítulo 4 para a codificação intra do padrão HEVC exploraram três correlações principais: (i) a correlação entre a orientação de borda dominante calculada a partir da textura da PU e as direções de predição intra; (ii) a correlação entre os modos intra escolhidos em PUs de níveis adjacentes da árvore de codificação; e (iii) a correlação entre os conjuntos de modos selecionados. A seguir, na Seção 5.1.1, visando a redução de complexidade da codificação multivista, as correlações entre as decisões de melhor modo intra em vistas vizinhas foram também exploradas.

Seguindo as explorações e análises de correlações entre vistas, a segunda contribuição mencionada na Seção 1.3.2, a qual visava a redução de complexidade das decisões de profundidade da árvore de codificação foi concretizada. Os métodos apresentados nas Seções 5.1.2 e 5.2.1 tratam dessa questão, explorando as correlações entre as estruturas das árvores de codificação de vistas adjacentes, e dando origem a otimizações nas decisões de profundidade máxima da árvore de CUs.

Ainda no cenário multivista, os métodos apresentados no Capítulo 5, Seções 5.2.2 e 5.2.3, concretizam a contribuição citada na Seção 1.3.3, a qual visava a redução de complexidade da codificação intra das vistas de textura e mapas de profundidade do codificador 3D-HEVC. Tais métodos, inicialmente, exploraram a redução de

complexidade da codificação intra dos mapas de profundidade através da análise de ocorrência dos novos modos DMMs adicionados a predição intra do 3D-HEVC; da correlação entre a escolha do modo Planar como o melhor modo intra e sua presença no conjunto de MPMs selecionados; e da identificação das bordas dominantes na textura das PUs.

Em seguida, expandindo a exploração da codificação intra dos mapas de profundidade para uma exploração que abordava também as vistas de textura utilizadas na codificação 3D-HEVC, um novo método de predição intra rápido foi proposto. Tal método realiza uma análise da textura extraíndo a informação de borda a partir dos pixels das PUs de textura e profundidade, utilizando o filtro de Sobel. Como resultado dessa filtragem, as amplitudes dos pixels com a mesma direção de modo intra são usadas para gerar um histograma de direções de modo. De forma que, os modos com os maiores valores de amplitude são selecionados para compor o subconjunto de modos candidatos que serão avaliados no cálculo RDO. Assim, a avaliação de todos os 33 modos intra angulares disponíveis já não é necessária, mas somente dos modos selecionados de acordo com o histograma de direções de modo (mais os modos Planar e DC), os quais serão testados durante a pesquisa RDO para encontrar o melhor modo de predição intra.

Os resultados experimentais demonstraram que considerando a redução de complexidade da codificação intra no cenário de vista única ou *Singleview*, os algoritmos de decisão de modo permitem alcançar uma redução de aproximadamente 40% em média, com uma perda mínima em desempenho de codificação de 0,08dB em BD-PSNR em média, quando comparados ao software de referência do HEVC. Estes resultados são os melhores em termos de redução de complexidade computacional entre os trabalhos relacionados, descritos na Seção 4.2. Além disso, considerando o método de predição inter-vista do modo intra proposto na Seção 5.1.1, uma redução de até 47,77% foi alcançada, com uma perda mínima em eficiência de codificação.

Com respeito aos algoritmos para a otimização da árvore de codificação usada durante a predição intra, inicialmente uma redução de cerca de 29% em média, foi alcançada quando comparado ao HEVC Simulcast. Quando utilizando o 3D-HEVC, uma redução no tempo de processamento de aproximadamente 28% em média foi

alcançada, com uma perda mínima em termos de eficiência de codificação de 0,06 dB em BD-PSNR em média.

Além disso, considerando as soluções propostas para a codificação intra das vistas de textura e mapas de profundidade do 3D-HEVC, a otimização da predição intra dos mapas de profundidade realizada inicialmente, alcançou uma redução de 15% no tempo de processamento de codificação intra da profundidade, com uma perda insignificante de 0,52% em eficiência de codificação, em média.

Logo após, a partir do método proposto para a codificação intra das vistas de textura e mapas de profundidade foi possível alcançar uma redução de 26%, em média, na complexidade global do codificador 3D-HEVC, com uma perda mínima em *BD-Rate* de 0,32% em média, para as vistas de textura, e de 0,45% em média, para as vistas sintetizadas.

Como trabalhos futuros, tendo em conta que o objetivo dos sistemas de vídeo multimídia é habilitar os dispositivos de consumo à atingir codificação em tempo real para vídeos de alta resolução, soluções para arquitetura de hardware devem ser estudadas.

Do ponto de vista de implementações em hardware, o filtro de Sobel é também uma boa escolha para a redução de complexidade da codificação intra do 3D-HEVC, uma vez que é mais fácil de implementar em arquiteturas de hardware dedicado do que outros filtros [115]. Os filtros de Sobel possuem um número reduzido de entradas (3x3), sendo menos sensíveis ao ruído do que outros detectores de borda [115][116]. Além disso, considerando que as máscaras de Sobel contêm apenas quatro multiplicações por 2, as quais podem ser realizadas eficientemente utilizando registros de deslocamento (*shift registers*), é possível implementar o filtro de Sobel em um circuito livre de multiplicadores, otimizando a arquitetura dedicada desenvolvida.

Ainda como trabalhos futuros, técnicas de otimização adicionais podem ser integradas ao método de redução da complexidade da codificação intra do 3D-HEVC apresentado na Seção 5.2.3 deste trabalho. Entre essas técnicas, soluções para a redução de complexidade da codificação dos DMMs, simplificando a determinação das partições Wedgelet; terminação antecipada das árvores de CUs das vistas de textura e mapas de

profundidade entre outras, a fim de reduzir ainda mais a complexidade computacional do codificador 3D-HEVC.

7. REFERÊNCIAS

- [1] ITU-T and ISO/IEC, “Advanced video coding for generic audio visual services,” ITU-T Recommendation H.264 and ISO/IEC 14496-10, Mar. 2005.
- [2] T. Wiegand, G. Sullivam, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [3] A. Puri, X. Chen, and A. Luthra, “Video coding using the H.264/MPEG-4 AVC compression standard,” *Signal Process. Image Commun.*, vol. 19, no. 9, pp. 793-849, Oct. 2004.
- [4] ITU-T and ISO/IEC JCT-VC, “High Efficiency Video Coding,” ITU-T Recommendation H.265 | International Standard, Apr. 2013.
- [5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [6] K. Ugur, K. R. Andersson, A. Fuldseth, and G. Bjøntegaard, “Low complexity video coding and the emerging HEVC standard,” in *Proc. of Picture Coding Symposium (PCS)*, Nagoya, Japan, p.474-477, Dec. 2010.
- [7] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang, “Joint Draft 8.0 on Multiview Video Coding Joint Video Team,” in *Document JVT-AB204, 28th JVT Meeting*, Hannover, Germany, Jul. 2008.
- [8] J. Konieczny and M. Domański, “Depth-Based Inter-View Motion Data Prediction for HEVC-Based Multiview Video Coding,” in *Proc. of Picture Coding Symposium (PCS)*, Kraków, Poland, pp. 33-36, 2012.
- [9] D. Tian, P. Pandit, P. Yin, and C. Gomila, “Study of MVC Coding Tools,” in *Document JVT-Y044, 26th JVT Meeting*, Shenzhen, China, Oct. 2007.
- [10] T. Chen, Y. Kashiwagi, C. Lim, and T. Nishi, “Coding performance of Stereo High Profile for movie sequences”, in *Document JVT-AE022, 32nd JVT Meeting*, London, United Kingdom, Jul. 2009.
- [11] G. Tech, K. Wegner, Y. Chen, and S. Yea, “Test Model 10 of 3D-HEVC and MV-HEVC,” in *Document JCT3V-J1003, 10th JCT-3V Meeting*, Strasbourg, Oct. 2014.

-
- [12] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366-3378, Sept. 2013.
- [13] G. J. Sullivan, J. M. Boyce, Y. Chen, J. -R. Ohm, C. A. Segall, and A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 1001-1016, Dec. 2013.
- [14] Q. Zhang, P. An, Y. Zhang, L. Shen, and Z. Zhang, "Low complexity multiview video plus depth coding," *IEEE Trans. Consumer Electron.*, vol. 57, no. 4, pp. 1857-1865, Nov. 2011.
- [15] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Proc. of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, San Jose, USA, pp. 93-104, May. 2004.
- [16] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient Compression of Multi-view Depth Data Based on MVC," in *Proc. of 3DTV Conference*, Kos Island, Greece, pp. 1-4, May. 2007.
- [17] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792-1801, Dec. 2012.
- [18] I. -K. Kim, J. Min, T. Lee, W. -J. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697-1706, Dec. 2012.
- [19] F. Bossen, B. Benjamin, S. Karsten, and D. Flynn, "HEVC Complexity and Implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685-1696, Dec. 2012.
- [20] U. K. Khan, M. Shafique, M. Grellert, and J. Henkel, "Hardware-Software Collaborative Complexity Reduction Scheme for the Emerging HEVC Intra Encoder," in *Proc. of IEEE/ACM 16th Design, Automation and Test in Europe (DATE)*, Grenoble, France, pp. 125-128, Mar. 2013.
- [21] G. Correa, P. Assunção, L. Agostini, and L. Cruz, "Performance and Computational Complexity Assessment of High Efficiency Video Encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1899-1909, Dec. 2012.
- [22] T. Silva, L. Agostini, and L. Cruz, "Fast HEVC Intra Prediction Mode Decision Based on Edge Direction Information," in *Proc. of IEEE European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, pp. 2014-2018, Aug. 2012.
-

-
- [23] T. Silva, L. Agostini, and L. Cruz, "Fast HEVC intra mode decision based on dominant edge evaluation and tree structure dependencies," in *Proc. of 19th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Seville, Spain, pp. 568-571, Dec. 2012.
- [24] T. Silva, L. Agostini, and L. Cruz, "HEVC intra coding acceleration based on tree inter-level mode correlation," in *Proc. of IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, Poznan, Poland, pp. 88-93, Sept. 2013.
- [25] T. Silva, L. Agostini, and L. Cruz, "HEVC intra mode decision acceleration based on tree depth levels relationship," in *Proc. of IEEE Picture Coding Symposium (PCS)*, San Jose, USA, pp. 277-280, Dec. 2013.
- [26] T. Silva, L. Agostini and, L. Cruz, "HEVC Intra Mode Decision Based on Prediction Unit Texture and Quad-tree Interlevel Correlation," in *Proc. of 9th Conference on Telecommunications (ConfTele)*, Castelo Branco, Portugal, pp. 229-231, May. 2013.
- [27] T. Silva, L. Agostini and, L. Cruz, "Speeding Up HEVC Intra Coding Based on Tree Depth Inter-levels Correlation Structure," in *Proc. of IEEE European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, pp. 1-5, Sept. 2013.
- [28] T. Silva, L. Agostini and, L. Cruz, "HEVC Intra Prediction Acceleration Based on Texture Direction and Prediction Unit Modes Reuse", *APSIPA Transactions on Signal and Information Processing*, Cambridge Journals, vol. 3, pp. 1-13, Dec. 2014.
- [29] T. Silva, L. Agostini and, L. Cruz, "Inter-view Prediction of Intra Mode Decision for High-Efficiency Video Coding-based Multiview Video Coding," *Journal of Electronic Imaging (JEI)*, vol. 23, no. 3, pp. 033008, Jun. 2014.
- [30] T. Silva, L. Agostini and, L. Cruz, "Inter-view Prediction of Coding Tree Depth for HEVC-based Multiview Video Coding," in *Proc. of IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Abu Dhabi, United Arab Emirates, pp.165-168, Dec. 2013.
- [31] T. Silva, L. Agostini and, L. Cruz, "Complexity Reduction of Depth Intra Coding for 3D Video Extension of HEVC," in *Proc. of IEEE Visual Communications and Image Processing (VCIP)*, Valletta, Malta, pp. 229-232, Dec. 2014.
- [32] Y. Lin, C. Kao, H. Kuo and, J. Chen, *VLSI Design for Video Coding: H.264/AVC Encoding from Standard Specification to Chip*, Springer: New York, 2010.
- [33] I. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed., John Wiley and Sons: Chichester, 2010.
-

-
- [34] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, The Institution of Electrical Engineers: London, 2003.
- [35] T. Xiaohua, L. Thinh and, L. Yong, *Entropy Coders of the H.264/AVC Standard: Algorithms and VLSI Architectures*, Springer-Verlag: Berlin, 2011.
- [36] J. Chen, U. Koc and, K. J. R. Liu, *Design of Digital Video Coding Systems: A Complete Compressed Domain Approach*, New Work: Marcel Dekker, 2002.
- [37] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley and Sons: Chichester, 2003.
- [38] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd ed., Kluwer Academic Publishers: Boston, 1997.
- [39] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publishers: Boston, 1999.
- [40] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," in *Document VCEG-M33, ITU-T SG16/Q6 VCEG*, Austin, Texas, Mar. 2001.
- [41] G. Bjøntegaard, "Improvements of the BD-PSNR model," in *Document VCEG-A111, ITU-T SG16/Q6 VCEG*, Berlin, Germany, Jul. 2008.
- [42] High Efficiency Video Coding Test Model. Available in: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-10.0/, Jun. 2014.
- [43] F. Bossen, "Common Test Conditions and Software Reference Configurations," in *Document JCTVC-L1100, 12th JCT-VC Meeting*, Geneva, Switzerland, Jan. 2013.
- [44] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan "High Efficiency Video Coding (HEVC) Test Model 16 Encoder Description," JCTVC-R1002, 18th JCT-VC Meeting, Sapporo, Japão, Jul. 2014.
- [45] P. Farrelle, *Recursive block coding for image data compression*, Springer-Verlag: New York, 1990.
- [46] T. Schierl, V. George, A. Henkel, and D. Marpe, "Dependent Slices," in *Document JCTVC-I0229, 9th JCT-VC Meeting*, Geneva, Switzerland, May, 2012.
- [47] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An Overview of Tiles in HEVC," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 969-977, Dec. 2013.
- [48] M. Zhou, W. Gao, M. Jiang, and H. Yu, "HEVC Lossless Coding and Improvements," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1839-1843, Dec. 2012.
-

-
- [49] K. Ugur, K. R. Andersson, and A. Fuldseth, "Description of video coding technology proposal by Tandberg, Nokia, Ericsson," in *Document JCTVC-A119, 1st JCT-VC Meeting*, Dresden, Germany, Apr. 2010.
- [50] I.-K. Kim et. al., "High Efficiency Video Coding (HEVC) Test Model 15 Encoder Description," in *Document JCTVC-Q1002, 17th JCT-VC Meeting*, Valencia, Spain, Apr. 2014.
- [51] D. Marpe, H. Schwarz, and T. Wiegand, "Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620-636, Jul. 2003.
- [52] D. Wood, "The truth about stereoscopic television", in *Proc. of International Broadcasting Convention (IBC)*, Amsterdam, Netherlands, pp 45-51, Sept. 2009.
- [53] J. Faubert, "Motion parallax, stereoscopy, and the perception of depth: Practical and theoretical issues", in *Proceedings of SPIE Three-dimensional video and display: devices and systems*, Boston, USA, pp. 168-191, Jan. 2001.
- [54] N. A. Dodgson, "Autostereoscopic 3D Displays", *IEEE Computer*, vol. 38, no. 8, pp. 31- 36, Aug. 2005.
- [55] L. Kong, G. Jin, X. Zhong, and F. Mechanics, "An autostereoscopic projecting system based on parallax barrier and lenticular sheets," in *Proc. of International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, pp. 4816-4819, Jul. 2011.
- [56] M. Tanimoto, "FTV: Free-viewpoint Television," *Signal Process. Image Commun.*, vol. 27, no. 6, pp. 555-570, Jul. 2012.
- [57] K. Muller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," in *Proc. of IEEE - Spec. Issue 3D Media Displays*, vol. 99, no. 4, pp. 643-656, Apr. 2011.
- [58] F. Zou, D. Tian, A. Vetro, H. Sun, O. Au, and S. Shimizu, "View Synthesis Prediction in the 3D Video Coding Extensions of AVC and HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol.24, no. 10, pp. 1696–1708, Oct. 2014.
- [59] C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto, *3D-TV System with Depth-Image-Based Rendering - Architectures, Techniques and Challenges*, Springer: New York, 2013.
- [60] H. Schwarz and T. Wiegand, "Inter-view prediction of motion data in multiview video coding," in *Proc. of Picture Coding Symposium (PCS)*, pp. 101-104, May 2012.
- [61] G. Tech, K. Wegner, Y. Chen, and S. Yea, "3D-HEVC Test Model 3", in *Document JCT3V-C1005, 3rd JCT-3V Meeting*, Geneva, Switzerland, Jan. 2013.
-

-
- [62] A. Vetro, Y. Chen, and A. Vetro, "Next-Generation 3D Formats with Depth Map Support," *IEEE Multimedia*, vol. 21, no. 2, pp. 90-94, Jun. 2014.
- [63] Y. Ho and Y. Song, "MPEG Activities for 3D Video Coding," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Angkor Wat, Cambodia, pp. 3-6, Dec. 2014.
- [64] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, M. Karsten, G. Tech, M. Winken, and T. Wiegand, "3D Video Coding Using Advanced Prediction, Depth Modeling, and Encoder Control Methods," in *Picture Coding Symposium*, Krakow, Poland, pp. 3-6, May. 2012.
- [65] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1668-1683, Dec. 2012.
- [66] V. Baroncini, G. J. Sullivan, and J. -R. Ohm, "Report on Preliminary Subjective Testing of HEVC Compression Capability," in *Document JCTVC-H1004, 8th JCT-VC Meeting*, San Jose, Feb. 2012.
- [67] B. Li, G. J. Sullivan, and J. Xu, "Comparison of Compression Performance of HEVC Draft 10 with AVC High Profile," in *Document JCTVC-M0329, 13th JCT-VC Meeting*, Incheon, South Korea, 1-3, Apr. 2013.
- [68] JCT-VC, "Test model under consideration," in *Document JCTVC-A205, 1st JCT-VC Meeting*, Dresden, Germany, Apr. 2010.
- [69] I. -K. Kim, K. McCann, K. Sugimoto, B. Bross, and W. -J. Han, "High Efficiency Video Coding (HEVC) Test Model 10 (HM10) Encoder Description," in *Document JCTVC-L1002, 12th JCT-VC Meeting*, Geneva, 1-36, Jan. 2013.
- [70] C. -W. Ku, C. -C. Cheng, G. -S. Yu, M. -C. Tsai, and T. -S. Chang, "A High-Definition H.264/AVC Intra-Frame Codec IP for Digital Video and Still Camera Applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 8, pp. 917-928, Aug. 2006.
- [71] E. M. Hung, R. L. Queiroz, and D. Mukherjee, "Inter-Frame Post-Processing for Intra-Coded Video," *Journal of Communication and Information Systems*, vol. 1, no. 28, pp. 1-7, Apr. 2013.
- [72] T. Wedi, H. Ohtaka, J. Wus, and S. Sekiguchi, "Intra-only H.264/AVC profiles for professional applications," in *Document JVT-U120, 21st JVT Meeting*, Hangzhou, 1-6, Oct. 2006.
- [73] G. J. Sullivan, Y. Haoping, S. -I. Sekiguchi, S. Huifang, T. Wedi, S. Wittmann, L. Yung-Lyul, A. Segall, and T. Suzuki, "New Standardized Extensions of MPEG4-
-

-
- AVC/H.264 for Professional-Quality Video Applications,” in *Proc. of International Conference on Image Processing (ICIP)*, San Antonio, USA, pp. 13-16, Sept. 2007.
- [74] High Efficiency Video Coding Test Model. Available in: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-2.0/, Jan. 2014.
- [75] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Process. Mag.*, vol. 15, pp. 23-50, Nov. 1998.
- [76] Y. Kim, D. Jun, S. -H. Jung, J. S. Choi, and J. Kim, “A Fast Intra-Prediction Method in HEVC Using Rate-Distortion Estimation Based on Hadamard Transform,” *ETRI Journal*, vol. 35, no. 2, pp. 270-280, Apr. 2013.
- [77] M. U. K. Khan, M. Shafique, M. Grellert, and J. Henkel, “Hardware-Software Collaborative Complexity Reduction Scheme for the Emerging HEVC Intra Encoder,” in *Proc. of IEEE/ACM 16th Design, Automation and Test in Europe (DATE)*, pp.125-128, Grenoble, France, Mar. 2013.
- [78] L. Zhao, L. Zhang, M. Siwei, and D. Zhao, “Fast Mode Decision Algorithm for Intra Prediction in HEVC”, in *IEEE Visual Communications and Image Processing (VCIP) Conference*, Tainan, Taiwan, pp. 1-4, Nov. 2011.
- [79] J. Kim, Y. Choe, and Y. -G. Kim, “Fast Coding Unit Size Decision Algorithm for Intra Coding in HEVC”, in *Proc. of IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, pp. 637-638, Jan. 2013.
- [80] W. Jiang, H. Ma, and Y. Chen, “Gradient Based Fast Mode Decision Algorithm for Intra Prediction in HEVC”, in *Proc. of 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, Yichang, China, pp. 1836-1840, Apr. 2012.
- [81] G. Tian, and S. Goto, “Content Adaptive Prediction Unit Size Decision Algorithm for HEVC Intra Coding”, in *Proc. of 29th Picture Coding Symposium (PCS)*, Krakow, Poland, pp. 405-408, May. 2012.
- [82] J. Kim, J. Yang, H. Lee, and B. Jeon, “Fast Intra Mode Decision of HEVC based on Hierarchical Structure”, in *Proc. of 8th International Conference on Information Communications and Signal Processing (ICICS)*, Singapore, pp. 1-4, Dec. 2011.
- [83] M. Zhang, C. Zhao, and J. Xu, “An Adaptive Fast Intra Mode Decision in HEVC”, in *Proc. of International Conference on Image Processing (ICIP)*, Orlando, USA, pp. 221-224, Sept. 2012.
-

-
- [84] H. Zhang and Z. Ma, “Fast Intra Prediction for High Efficiency Video Coding”, in *Proc. of Pacific-Rim Conference on Multimedia (PCM)*, Singapore, pp. 1-10, Dec. 2012.
- [85] “MPEG-7 XM Document: MPEG-7 Visual Part Experimentation Model Version 10.0,” in *Document N4063, ISO/IEC/JTC1/SC29/WG11 MPEG*, Singapore, Mar. 2001.
- [86] K. Müller and A. Vetro, “Common Test Conditions of 3DV Core Experiments”, in *Document JCT3V-G1100, 7th JCT-3V Meeting*, San José, US, Jan. 2014.
- [87] P. Merkle, K. Müller, A. Smolic, and T. Wiegand, “Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG-AVC,” in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, Toronto, Canada, pp.1717-1720, Jul. 2006.
- [88] M. Fierl and B. Girod, “Multiview Video Compression,” *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 66-76, Nov. 2007.
- [89] K. Wegner, O. Stankiewicz, K. Klimaszewski, and M. Domanski, “Comparison of multiview compression performance using MPEG-4 MVC and prospective HEVC technology,” in *Document M17913, ISO/IEC JTC1/SC29/WG11 MPEG*, Geneva, Switzerland, Jul. 2010.
- [90] J.-R. Ohm, “Overview of 3D Video Coding Standardization,” in *Proc. of Three Dimensional Systems and Applications (3DSA)*, Keynote Speech 2, Osaka, Japan, Jun. 2013.
- [91] Multiview extension of High Efficiency Video Coding Test Model. Available in: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-12.1/ Fev. 2014.
- [92] Y. P. Su, A. Vetro, and A. Smolic, “Common Test Conditions for Multiview Video Coding,” in *Document JVT-U211, 21st JVT Meeting*, Hangzhou, China, Oct. 2006.
- [93] G. J. Sullivan and R. L. Baker, “Efficient quadtree coding of images and video,” *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 327-31, Jan. 1994.
- [94] C. Zhou, F. Zhou, and Y. Chen, “Spatio-temporal correlation-based fast coding unit depth decision for high efficiency video coding,” *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 1-13, Aug. 2013.
- [95] Y. Zhang, H. Wang, and Z. Li, “Fast Coding Unit Depth Decision Algorithm for Interframe Coding in HEVC,” in *Proc. of Data Compression Conference (DCC)*, Snowbird, USA, pp. 54-62, Mar. 2013.
-

-
- [96] G. Correa, P. Assunção, L. Agostini, and L. Cruz, "Complexity Control of High Efficiency Video Encoders for Power-Constrained Devices," *IEEE Trans. on Consumer Electron.*, vol. 57, no. 4, pp. 1866-1874, Nov. 2011.
- [97] R. Fan, Y. Zhang, Z. Li, and N. Wang, "An Improved Similarity-Based Fast Coding Unit Depth Decision Algorithm for Inter-frame Coding in HEVC," *MultiMedia Modeling, Lecture Notes in Computer Science*, vol. 8325, pp. 529-540, Jan. 2014.
- [98] G. Correa, P. Assunção, L. Agostini, and L. Cruz, "Fast HEVC Encoding Decisions Using Data Mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no.99, pp. 1-13, 2014.
- [99] H. S. Lee, K. Y. Kim, T. R. Kim, and G. H. Park, "Fast encoding algorithm based on depth of coding unit for high efficiency video coding," *Optical Engineering*, vol. 51, no. 6, pp. 1-11, May. 2012.
- [100] Q. Yu, X. Zhang, S. Wang and S. Ma, "Early Termination of Coding Unit Splitting for HEVC," in *Proc. of Annual Summit and Conference Asia Pacific Signal and Information Processing Association (APSIPA ASC)*, Hollywood, USA, pp.1-4, Dec. 2012.
- [101] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465-470, Feb. 2013.
- [102] Mobile 3DTV content delivery optimization over DVB-H system, 3D Video Database. Available in: <http://sp.cs.tut.fi/mobile3dtv/stereo-video/>, Mar. 2014.
- [103] K. Muller, K. Smolic, P. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View Synthesis for Advanced 3D Video Systems," *EURASIP Journal on Image Video Processing, Special Issue 3D Image Video Processing*, vol. 2008, no. 438148, pp. 1-11, Feb. 2009.
- [104] I. Sobel, *Machine Vision for Three-Dimensional Scenes*, H. Freeman (editor). Academic Press: New York, 1990.
- [105] O. R. Vincent and O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection", in *Proc. of Proc. of Informing Science & IT Education Conference (InSITE)*, Macon, USA, pp.97-107, May. 2009.
- [106] R. Maini and H. Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques", *I. J. of Image Processing (IJIP)*, vol. 3, no. 1, pp. 1-12, Feb. 2009.
-

-
- [107] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches”, *Int. J. of Recent Trends in Engineering and Technology*, vol. 1, no. 2, pp. 250-254, May. 2009.
- [108] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall: Upper Saddle River, 2002.
- [109] E. G. Mora, J. Jung, M. Cagnazzo, S. Member, and B. Pesquet-popescu, “Initialization , limitation and predictive coding of the depth and texture quadtree in 3D-HEVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1554-1565, Sept. 2014.
- [110] E. G. Mora, J. Jung, M. Cagnazzo, and B. Pesquet-Popescu, “Depth video coding based on intra mode inheritance from texture,” *APSIPA Trans. Signal Inf. Process.*, vol. 3, pp.1-13, Jan. 2014.
- [111] Z. Gu, J. Zheng, N. Ling, and P. Zhang, “Fast Intra Prediction Mode Selection for Intra Depth Map Coding,” in *Document JCT3V-E0238, 5th JCT-3V Meeting*, Vienna, Austria, Aug. 2013.
- [112] G. Sanchez, M. Saldanha, G. Balota, B. Zatt, M. Porto, and L. Agostini, “Complexity Reduction for 3D-HEVC Depth Maps Intra-frame Prediction using Smart Edge Detector Algorithm”, in *Proc. of IEEE International Conference on Image Processing (ICIP)*, Paris, France, pp. 3209-3213, Oct. 2014.
- [113] C. Park, “Edge-Based Intramode Selection for Depth-Map Coding in 3D-HEVC,” *IEEE Trans. Image Process.*, vol. 24, no. 1, pp 155-162, Nov. 2014.
- [114] Z. Gu, J. Zheng, N. Ling, and P. Zhang, “Fast Depth Modeling Mode Selection for 3D HEVC Depth Intra Coding,” in *Proc. of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, San Jose, USA, pp. 1-4, Jul. 2013.
- [115] Z. E. M. Osman, F. A. Hussin, and N. B. Z. Ali, “Optimization of Processor Architecture for Image Edge Detection Filter,” in *Proc. of 12th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom, pp. 648–652, Mar. 2010.
- [116] T. A. Abbasi and M. U. Abbasi, “A novel FPGA-based architecture for Sobel edge detection operator,” *Int. J. of Electronics*, vol. 94, no. 9, pp. 889-896, Sept. 2007.
-

APÊNDICE A: LISTA DE ARTIGOS CIENTÍFICOS PUBLICADOS E COMUNICAÇÕES PROFERIDAS

Artigos em Conferências Nacionais e Internacionais:

- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Inter-view Prediction of Coding Tree Depth for HEVC-based Multiview Video Coding. In: *IEEE International Conference on Multimedia and Expo (ICME 2015)*, Turin, 2015 (artigo aceito para publicação).
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Complexity Reduction of Depth Intra Coding for 3D Video Extension of HEVC. In: *IEEE Visual Communications and Image Processing Conference (VCIP 2014)*, Malta, pp. 229 - 232, 2014.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. HEVC intra coding acceleration based on tree inter-level mode correlation. In: *IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA 2013)*, Poznan, pp.88 - 93, 2013.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. HEVC intra mode decision acceleration based on tree depth levels relationship. In: *IEEE Picture Coding Symposium (PCS 2013)*, San Jose, pp. 277 - 280, 2013.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Inter-view Prediction of Coding Tree Depth for HEVC-based Multiview Video Coding. In: *IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2013)*, Abu Dhabi, pp. 165 - 168, 2013.

- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. HEVC Intra Mode Decision Based on Prediction Unit Texture and Quad-tree Interlevel Correlation. In: *9th Conference on Telecommunications (ConfTele 2013)*, Castelo Branco, pp.229 - 231, 2013.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Speeding Up HEVC Intra Coding Based on Tree Depth Inter-levels Correlation Structure. In: *IEEE European Signal Processing Conference (EUSIPCO 2013)*, Marrakech, pp. 1 - 5, 2013.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Fast HEVC intra mode decision based on dominant edge evaluation and tree structure dependencies. In: *IEEE International Conference on Electronics, Circuits and Systems (ICECS 2012)*, Seville, pp. 568 - 571, 2012.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Fast HEVC Intra Prediction Mode Decision Based on Edge Direction Information. In: *IEEE European Signal Processing Conference (EUSIPCO 2012)*, Bucareste, pp.2014 - 2018, 2012.

Artigos em Revistas Internacionais:

- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Inter-view Prediction of Intra Mode Decision for High-Efficiency Video Coding-based Multiview Video Coding. *Journal of Electronic Imaging (JEI)*, v.23, pp. 033008, 2014.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. HEVC Intra Prediction Acceleration Based on Texture Direction and Prediction Unit Modes Reuse. *APSIPA Transactions on Signal and Information Processing*, Cambridge Journals, v.3, pp. 1-13, 2014.
- SILVA, Thaísa Leal da, AGOSTINI, Luciano Volcan, CRUZ, Luis Alberto da Silva. Fast Intra Prediction Algorithm Based on Texture Analysis for 3D-

HEVC Encoders. *Journal of Real-Time Image Processing*, 2015 (artigo submetido).

APÊNDICE B: CENÁRIO DAS SIMULAÇÕES - CONDIÇÕES COMUNS DE TESTE

Este apêndice apresenta as condições comuns teste (*CTCs - Common Test Conditions*) [86] usadas no decorrer deste trabalho, bem como os cenários de testes e as características das sequências de vídeo usadas.

As *CTCs* incluem características que devem ser consideradas durante a codificação dos vídeos, como os conjuntos de parâmetros de quantização (QPs) a serem utilizados para codificar as vistas de textura e os mapas de profundidade. Além disso, as *CTCs* indicam algumas sequências de vídeo com características diversas para serem utilizadas durante as simulações de teste.

Três cenários de teste foram considerados nesse trabalho: codificação de vista única, codificação multivista sem mapas de profundidade (somente vistas de textura), e codificação multivista com mapas de profundidade. De modo que, nos testes com codificação de vista única, apenas a primeira vista de cada sequência multivista foi utilizada nas simulações; e nos testes com codificação multivista, três vistas foram utilizadas.

As sequências de teste recomendadas nas *CTCs* são apresentadas na Tabela B.1. De forma que, a ordem de codificação das três vistas de cada sequência (apresentadas na coluna "3 Vistas") é, primeiramente, a vista central, depois a vista da esquerda e por último a da direita.

Tabela B.1: Características das sequências de teste das *CTCs*.

Sequência	Taxa de Tramas (<i>fps</i>)	Número de Tramas	3 Vistas	Resolução (<i>pixels</i>)
Balloons	30	300	1-3-5	1024x768
Newspaper	30	300	1-3-5	1024x768
Kendo	30	300	2-4-6	1024x768
GTFly	25	250	9-5-1	1920x1080
PoznanStreet	25	250	5-4-3	1920x1080
UndoDancer	25	250	1-5-9	1920x1080
PoznanHall2	25	200	7-6-5	1920x1080
Shark	30	300	1-5-9	1920x1080

A seguir são apresentadas algumas das principais características de cada uma das sequências de teste recomendadas pelas CTCs.

A sequência *Balloons* foi filmada em ambiente interno, em estúdio. Tal sequência apresenta movimento de câmera, e o movimento de objetos é complexo com reflexões e transparência. O nível de detalhes da textura é médio, e o nível de complexidade da estrutura de profundidade também. A sequência foi filmada com luz de estúdio conforme descrito em [102]. A Figura B.1 (a) e a Figura B.1 (b), apresentam uma trama de textura da sequência *Balloons* e seu respectivo mapa de profundidade.

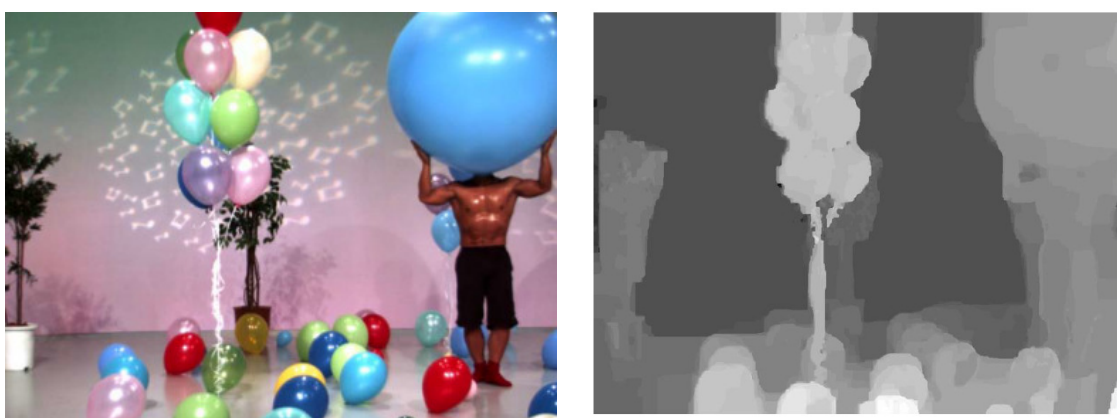


Figura B.1: Trama de textura (a) e Trama de profundidade (b) da sequência *Balloons*.

Conforme mencionado em [102], a sequência *Newspaper* foi filmada em ambiente interno, em estúdio. Tal sequência não apresenta movimento de câmera (câmera estática), e o movimento de objetos é simples. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade é médio, a sequência foi feita com luz de estúdio [102]. A Figura B.2 (a) e Figura B.2 (b) apresentam uma trama de textura da sequência *Newspaper* e seu mapa de profundidade, respectivamente.

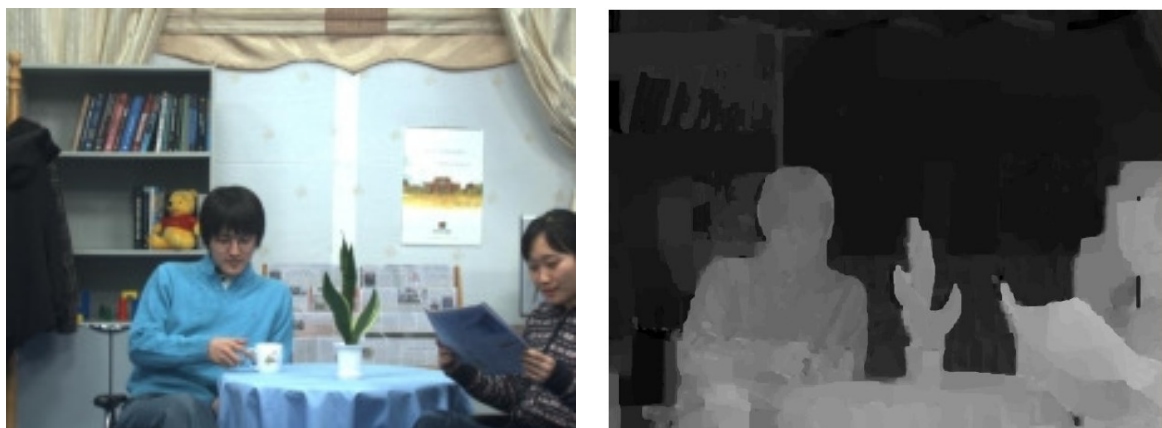


Figura B.2: Trama de textura (a) e Trama de profundidade (b) da sequência Newspaper.

A sequência Kendo foi filmada em ambiente interno, em estúdio. Essa sequência apresenta movimento de câmera, e o movimento de objetos é complexo, apresentando reflexões, névoa e transparência, conforme descrição apresentada em [102]. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade é médio. A sequência foi filmada utilizando luz de estúdio [102]. A Figura B.3 (a) e a Figura B.3 (b) apresentam uma trama de textura da sequência Kendo e mapa de profundidade associado, respectivamente.



Figura B.3: Trama de textura (a) e Trama de profundidade (b) da sequência Kendo.

A sequência GTFly foi produzida utilizando computação gráfica. Tal sequência não apresenta movimentação de objetos complexa. A textura apresenta um nível de detalhamento médio, e o nível de complexidade da estrutura de profundidade também é médio. As Figuras B.4 e B.5, apresentam uma trama de textura da sequência GTFly e seu respectivo mapa de profundidade.

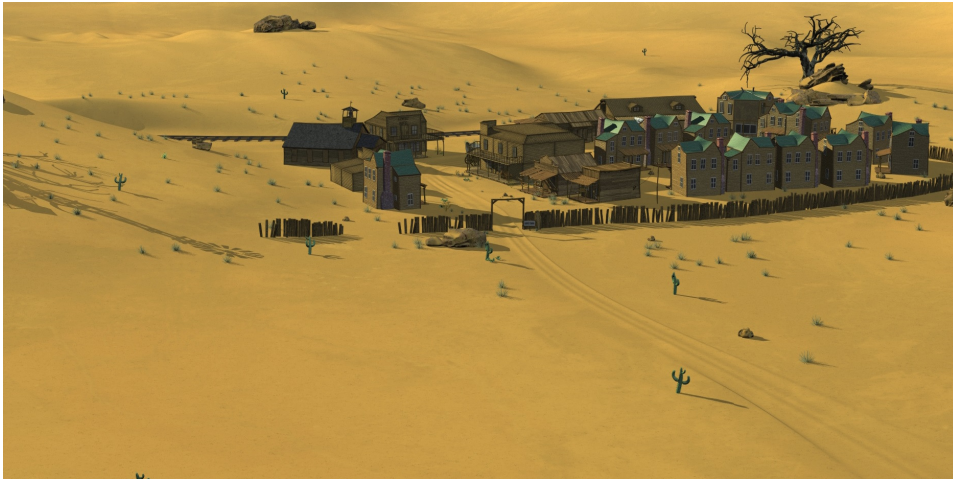


Figura B.4: Trama de textura da sequência GTFly.



Figura B.5: Trama de profundidade da sequência GTFly.

A partir da descrição do ambiente de filmagem da sequência PoznanStreet apresentada em [102]. A sequência foi filmada em ambiente externo, utilizando luz natural. Tal sequência não apresenta movimento de câmera, e o movimento de objetos é classificado como complexo, apresentando reflexões e transparência. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade também. A Figura B.6 e a Figura B.7 apresentam uma trama de textura da sequência PoznanStreet e seu mapa de profundidade correspondente.



Figura B.6: Trama de textura da sequência PoznanStreet.

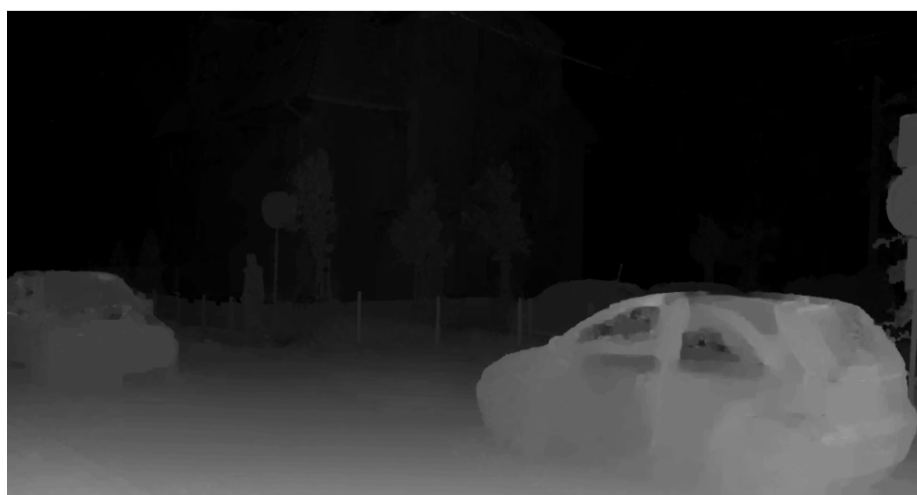


Figura B.7: Trama de profundidade da sequência PoznanStreet.

A sequência UndoDancer foi criada utilizando computação gráfica. Essa sequência apresenta movimento de câmera, e a movimentação de objetos pode ser classificada como simples. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade é baixo. A Figura B.8 e a Figura B.9, apresentam uma trama de textura da sequência UndoDancer e seu mapa de profundidade correspondente.



Figura B.8: Trama de textura da sequência UndoDancer.

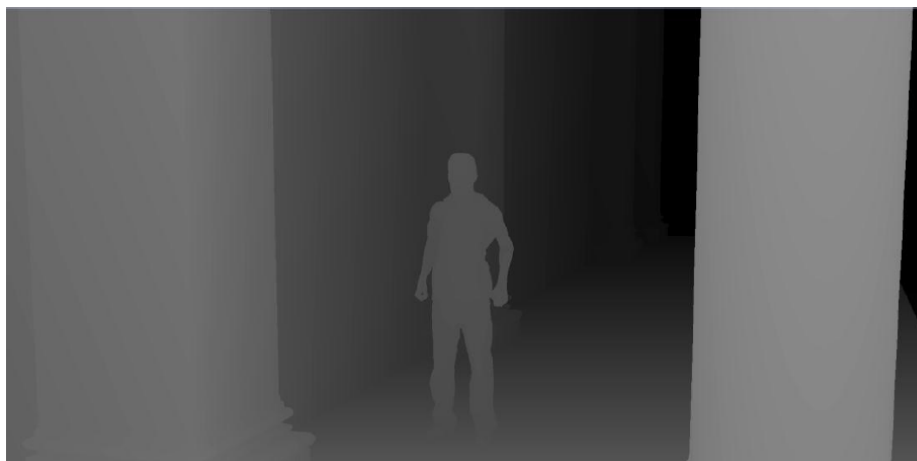


Figura B.9: Trama de profundidade da sequência UndoDancer.

De acordo com as informações apresentadas em [102], a sequência PoznanHall2 foi filmada em ambiente interno. Tal sequência apresenta movimento de câmera, e o movimento de objetos é complexo, apresentando reflexões e transparência. O nível de detalhes da textura é médio, e o nível de complexidade da estrutura de profundidade também. A sequência foi gravada utilizando luz natural [102]. A Figura B.10 e a Figura B.11, apresentam uma trama de textura da sequência PoznanHall2 e seu mapa de profundidade correspondente.

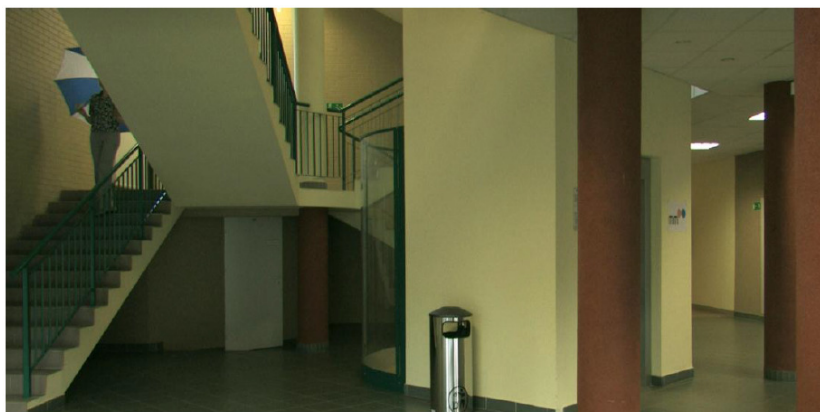


Figura B.10: Trama de textura da sequência PoznanHall2.



Figura B.11: Trama de profundidade da sequência PoznanHall2.

A sequência Shark foi produzida utilizando computação gráfica. Essa sequência apresenta movimento de câmera, e a movimentação de objetos é complexa. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade também. A Figura B.12 e a Figura B.13, apresentam uma trama de textura da sequência Shark e seu mapa de profundidade correspondente.



Figura B.12: Trama de textura da sequência Shark.



Figura B.13: Trama de profundidade da sequência Shark.

Além das oito sequências de teste especificadas em [86] e descritas acima, as sequências de teste: PoznanCarPark e PoznanHall1, também foram usadas neste trabalho para realização de análises estatísticas, as quais possuem resolução de 1920x1088 pixels e estão especificadas em [102]. Tais sequências são descritas em mais detalhe a seguir.

A sequência PoznanCarPark foi filmada em ambiente externo. Essa sequência não apresenta movimento de câmera (câmera estática), e o movimento de objetos é complexo, apresentando reflexões e transparência. O nível de detalhes da textura é alto, e o nível de complexidade da estrutura de profundidade também. A sequência foi filmada utilizando luz natural [102]. A Figura B.14 e a Figura B.15, apresentam uma trama de textura da sequência PoznanCarPark e seu mapa de profundidade correspondente.



Figura B.14: Trama de textura da sequência PoznanCarPark.



Figura B.15: Trama de profundidade da sequência PoznanCarPark.

A sequência PoznanHall1 foi filmada em ambiente interno e apresenta movimento de câmera. O movimento de objetos é complexo, usando reflexões e

transparência. O nível de detalhes da textura é médio, e o nível de complexidade da estrutura de profundidade também. A sequência foi filmada utilizando luz natural [102]. A Figura B.16 e a Figura B.17, apresentam uma trama de textura da sequência PoznanHall1 e seu mapa de profundidade associado.



Figura B.16: Trama de textura da sequência PoznanHall1.



Figura B.17: Trama de profundidade da sequência PoznanHall1.

Os valores de QPs indicados para codificação independente das vistas de textura são: 25, 30, 35 e 40. E os valores de QPs utilizados para a codificação dos mapas de profundidade são: 34, 39, 42 e 45 [86].