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

Dissertação

Redução do Tempo de Codificação de Mapas de Profundidade do 3D-HEVC Usando Árvores de Decisão Estáticas Construídas Através de *Data Mining*

Mário Roberto de Freitas Saldanha

Pelotas, 2018

Mário Roberto de Freitas Saldanha

Redução do Tempo de Codificação de Mapas de Profundidade do 3D-HEVC Usando Árvores de Decisão Estáticas Construídas Através de *Data Mining*

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

Orientador: Prof. Dr. Luciano Volcan Agostini
Coorientadores: Prof. Dr. César Augusto Missio Marcon
Prof. Dr. Marcelo Schiavon Porto

Pelotas, 2018

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S162r Saldanha, Mário Roberto de Freitas

Redução do tempo de codificação de mapas de profundidade do 3D-HEVC usando árvores de decisão estáticas construídas através de Data Mining / Mário Roberto de Freitas Saldanha ; Luciano Volcan Agostini, orientador ; César Augusto Missio Marcon, Marcelo Schiavon Porto, coorientadores. — Pelotas, 2018.

101 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2018.

1. Codificação de vídeo 3D. 2. 3D-HEVC. 3. Redução do tempo de codificação. 4. Mapas de profundidade. 5. Mineração de dados. I. Agostini, Luciano Volcan, orient. II. Marcon, César Augusto Missio, coorient. III. Porto, Marcelo Schiavon, coorient. IV. Título.

CDD : 005

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

AGRADECIMENTOS

Primeiramente gostaria de agradecer minha família pelo apoio incondicional durante todas as etapas da minha vida, em especial a minha mãe Adriani de Freitas Saldanha e meu pai Luis Pedro Saldanha, que foram essenciais para mais uma conquista. A eles agradeço todos os momentos que estavam disponíveis para me transmitir conhecimento e me ajudar a superar todos desafios que foram enfrentados ao longo destes 24 anos.

Também gostaria de agradecer aos amigos e orientadores Luciano Agostini, Marcelo Porto e César Marcon por compartilharem os conhecimentos, por auxiliar no desenvolvimento dos trabalhos para alcançar melhores resultados e permitirem que trabalhos de ótima qualidade tenham sido concluídos. Agradeço por me permitirem a fazer parte deste grupo qualificado e que possui um excelente ambiente de trabalho.

Agradeço ao amigo Gustavo Sanchez por continuarmos nessa parceria de trabalho que criamos desde o tempo que eu estava na graduação. Um grande amigo que também foi essencial para o desenvolvimento deste trabalho.

Ao pessoal do laboratório do GACI que sempre estiveram disponíveis para conversas ou para auxiliar com alguma dúvida. Sem esquecer do futsal semanal do GACI, que servia como um ótimo momento para descontrair.

Por fim, agradeço aos membros da banca por todas sugestões e críticas para melhorar este trabalho.

Obrigado a todos!

“*Imagine uma nova história para sua vida e acredite nela.”*
– Paulo Coelho

Resumo

Saldanha, Mário Roberto de Freitas. **Redução do Tempo de Codificação de Mapas de Profundidade do 3D-HEVC Usando Árvores de Decisão Estáticas Construídas Através de *Data Mining***. 2018. 101f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2018.

Esta dissertação apresenta uma solução para redução do tempo de codificação dos mapas de profundidade no 3D-*High Efficiency Video Coding* (3D-HEVC). Com a inserção dos mapas de profundidade no 3D-HEVC é possível reduzir significativamente o tamanho do vídeo codificado e transmistido para o decodificador. No entanto, os dados dos mapas de profundidade devem ser codificados de forma eficiente para que seja possível gerar as vistas sintetizadas com boa qualidade visual. Além das ferramentas utilizadas para codificar os dados de textura, o 3D-HEVC adiciona novas ferramentas desenvolvidas para codificação dos mapas de profundidade avaliando diversos tamanhos de blocos e modos de codificação, e isso gera um alto custo computacional. Este trabalho propõe uma abordagem com a utilização da mineração de dados para treinar seis árvores de decisão com informações extraídas do software de referência 3D-HEVC *Test Model* (3D-HTM) 16.0. Cada árvore de decisão é responsável por decidir se a Unidade de Codificação (UC) que está sendo codificada deve ser dividida em tamanhos menores. As árvores de decisão foram construídas para tamanhos de UCs 64×64, 32×32 e 16×16 e três árvores de decisão são especializadas para quadros I e três especializadas para quadros P e B. Avaliando a solução com o 3D-HTM 16.0 foi possível alcançar uma redução no tempo total de execução de 52% com um impacto desprezível de 0,18% considerando o *Bjontegaard Delta-rate* (*BD-rate*), quando comparado ao 3D-HTM sem modificações. Além disso, os resultados demonstraram que a solução supera os resultados alcançados pelos trabalhos relacionados.

Palavras-chave: codificação de vídeo 3D; 3D-HEVC; redução do tempo de codificação; mapas de profundidade; mineração de dados

Abstract

Saldanha, Mário Roberto de Freitas. **Time Reduction on 3D-HEVC Depth Maps Coding using Static Decision Trees Built Through Data Mining.** 2018. 101f. Dissertation (Master Degree) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2018.

This dissertation presents a solution for time reduction in 3D-High Efficiency Video Coding (3D-HEVC) depth maps coding. With the insertion of depth maps in 3D-HEVC was possible significantly reduce the size of the encoded and transmitted video. However, the depth maps should be efficiently encoded for generating synthesized views with good visual quality. In addition to the tools used to encode texture data, the 3D-HEVC adds new tools developed for encoding depth maps evaluating many block sizes and encoding modes generating a high computational cost. This work proposes an approach, which uses data mining technique for training six decision trees with information extracted from reference software 3D-HEVC Test Model (3D-HTM) 16.0. Each decision tree is responsible to decide if the evaluated Coding Unit (CU) should be divided into smaller sizes. The decision trees were constructed for Coding Units (CUs) sizes 64×64, 32×32 and 16×16 and three decision trees are specialized for I-frames and three specialized for P- and B-frames. When evaluating the solution with 3D-HTM 16.0 was possible to save 52% of total execution time with a negligible impact in Bjontegaard Delta-rate (*BD-rate*) of 0.18%, when compared to 3D-HTM without modifications. Besides, the results demonstrated that the solution outperformed the related works.

Key-words: 3D video coding; 3D-HEVC; coding time reduction; depth maps; data mining

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	All-Intra
AU	Access Unit
AVC	Advanced Video Coding
ARP	Advanced Residual Prediction
BD-Rate	Bjontegaard Delta Rate
CCT	Condições Comuns de Teste
CPV	Constant Partition Value
CTU	Coding Tree Unit
UC	Unidade de Codificação
DBBP	Depth Based Block Partitioning
DCP	Disparity-Compensated Prediction
DF	Deblocking Filter
DIS	Depth Intra Skip
DM	Data Mining
DMM	Depth Modeling Mode
DIBR	Depth-image-based rendering
DLT	Depth Lookup Table
FME	Fractional Motion Estimation
FPS	Frame per Second
GOP	Group of Pictures
HD	High Definition
HEVC	High Efficiency Video Coding
IC	Compensação de Iluminação
IG	Information Gain
IMP	Inter-view Motion Prediction
JCT-VC	Joint Collaborative Team on Video Coding
JCT-3V	Joint Collaborative Team on 3D Video Coding Extension Development
MCP	Motion-Compensated Prediction
ME	Motion Estimation
MPI	Motion Parameter Inheritance
MPM	Most Probable Mode
MVC	Multi-view Video Coding
MVD	Multi-View Plus Depth
PU	Prediction Unit
QP	Quantization Parameter
QTL	Quadtree Limitation
QTPred	Quadtree Prediction
RA	Random Access
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
REP	Reduced Error Pruning
RMD	Rough Mode Decision
SAO	Sample Adaptive Offset
SAD	Sum of Absolute Differences
SATD	Sum of Absolute Transformed Differences

SDC	Segment-wise Direct Component
SVDC	Synthesized View Distortion Change
VSO	View Synthesis Optimization
WEKA	Waikato Environment for Knowledge Analysis
2D	Two-Dimensional
3D	Three-Dimensional
3D-HEVC	3D-High Efficiency Video Coding
3D-HTM	3D-HEVC Test Model

SUMÁRIO

1 INTRODUÇÃO

Vídeos 3D trazem uma nova experiência visual para os espectadores transmitindo uma percepção de profundidade da cena, permitindo que os usuários desfrutem de uma experiência visual mais interessante do que a apresentada nos vídeos 2D (YASAKETHU, 2009). Com o crescimento do mercado de equipamentos que gravam e reproduzem vídeos 3D e o aumento nas resoluções de vídeos suportadas por estes equipamentos, existe uma grande demanda por soluções capazes de fornecer taxas de compressão elevadas mantendo alta qualidade visual.

Vídeos 3D são capturados de diferentes pontos de vistas simultaneamente. Estes pontos de vistas podem ser de diferentes câmeras ou uma única câmera com diferentes lentes para gravação do cenário. Uma vista consiste em um conjunto de quadros capturados por um determinado ponto de vista, também chamada de vista de textura (imagem colorida exibida para os visualizadores), onde cada quadro representa um instante de tempo da cena. Os vídeos 3D são construídos utilizando múltiplas vistas, criando a necessidade de manipular um conjunto de vídeos simultaneamente. Este conjunto de vídeos resulta em uma grande quantidade de dados gerados e, consequentemente, ocasiona um aumento considerável na largura de banda necessária para transmissão e no custo computacional de codificação para vídeos 3D.

Um codificador 3D deve ser capaz de receber dados de entrada de diferentes pontos de vistas, frequentemente de diferentes câmeras, e realizar a codificação de forma que as sequências de bits geradas na saída do codificador (*bitstream*) possam ser decodificadas por (i) um decodificador 2D (neste caso considera-se apenas uma vista), (ii) um decodificador de vistas estéreo (considera duas vistas), e/ou (iii) um decodificador de vistas 3D (pode considerar uma grande quantidade de vistas) (CHEN, 2008).

Vídeos 2D com resolução em alta definição já geram um grande desafio no processo de armazenamento e transmissão de vídeos em tempo real. Quando consideramos a manipulação de vídeos 3D, este processo torna-se ainda mais difícil, visto que estes vídeos utilizam uma maior quantidade de dados para sua representação.

Um vídeo 2D sem nenhuma técnica de compressão, capturado durante 5 minutos com uma taxa de amostragem de 30 quadros por segundo, resolução *High*

Definition 1080p (HD - 1920×1080 pixels), e representado por amostras de 8 bits e taxa de subamostragem de 4:2:0, requer aproximadamente 28 GB de espaço para armazenamento e uma largura de banda de 0,75 Gbps para transmissão em tempo real. Um vídeo 3D com 9 vistas de textura sem codificação, e as demais características iguais ao vídeo 2D, necessita de 252 GB de espaço para armazenamento e uma largura de banda de 6,75 Gbps para transmissão em tempo real.

A codificação de vídeo 2D visa reduzir os dados utilizados nos vídeos para representar a informação visual, explorando similaridades entre as informações presentes dentro do próprio quadro (redundância espacial) e entre quadros em diferentes instantes de tempo (redundância temporal). Além disso, também explora a probabilidade de ocorrências dos símbolos codificados (redundância entrópica) (RICHARDSON, 2010).

A redundância espacial refere-se às correlações dos pixels distribuídos espacialmente dentro do mesmo quadro de um vídeo e neste caso é utilizada a predição intra-quadro (LAINEMA, 2012). As redundâncias presentes entre quadros em diferentes instantes de tempo são codificadas pela predição inter-quadros, que explora a correlação entre quadros temporalmente próximos em um vídeo (GHANBARI, 2003). A redundância entrópica explora as probabilidades de ocorrências dos símbolos codificados. Basicamente, códigos com um número menor de bits são reservados para representar os símbolos com maior ocorrência e os símbolos com menor ocorrência utilizam códigos com uma maior quantidade de bits.

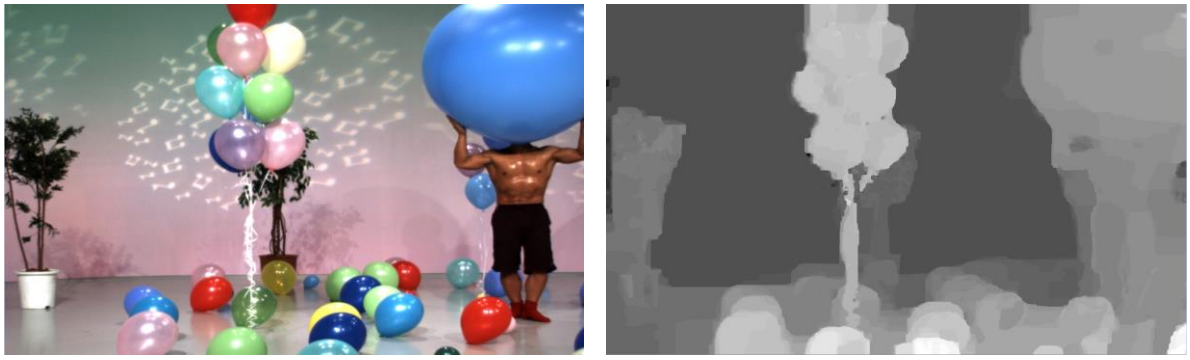
Todas essas ferramentas citadas para codificação de vídeos 2D também são utilizadas na codificação de vídeos 3D e, além disso, são utilizadas ferramentas para reduzir a quantidade de dados usados para representar as informações entre as diferentes vistas, o que é realizado pela predição inter-vistas, também conhecida como predição de disparidade (GUO, 2005).

Como mencionado anteriormente, um dos maiores desafios em vídeos 3D é a quantidade de dados gerados para transmissão e armazenamento, devido à necessidade de captura de mais de um ponto de vista. Considerando sistemas como *simulcast* (TECH, 2015), a quantidade de dados tende a aumentar proporcionalmente com o número de câmeras/vistas utilizadas durante a captura das cenas.

Com o objetivo de desenvolver um padrão avançado de codificação de vídeo

3D, capaz de alcançar elevadas taxas de compressão, o 3D-*High Efficiency Video Coding* (3D-HEVC) (MULLER, 2013) foi desenvolvido pelo *Joint Collaborative Team on 3D Video Coding Extension Development* (JCT-3V) (JCT-3V, 2016) como uma extensão para vídeos 3D do padrão de codificação de vídeos 2D *High Efficiency Video Coding* (HEVC) (SULLIVAN, 2012).

Visando aumentar a eficiência de codificação, o 3D-HEVC adota o formato de dados *Multi-View plus Depth* (MVD) (MERKLE, 2007), onde cada quadro de textura está associado a um quadro de mapa de profundidade correspondente. A Figura 1 (a) apresenta um exemplo de quadro de textura e seu mapa de profundidade associado para a sequência de vídeo 3D *Balloons* (MULLER, 2014). Como é possível observar na Figura 1 (b), os mapas de profundidade são representados por imagens em tons de cinza. Estas imagens são responsáveis por fornecer informações geométricas da cena, indicando a distância entre a câmera e um objeto.



(a) (b)

Figura 1 – (a) Quadro de textura e (b) seu mapa de profundidade associado.

Os mapas de profundidade são representados por amostras de oito bits com valores entre zero e 255 (como uma imagem em escala de tons de cinza), que representam a distância entre os objetos e a câmera.

Nos mapas de profundidade os objetos mais distantes da câmera são representados por tons de cinza mais escuro, onde os valores das amostras aproximam-se de zero, enquanto os objetos mais próximos da câmera são representados por tons de cinza mais claro, assumindo valores próximos de 255. Mapas de profundidade permitem definir claramente a presença de objetos presentes na cena, devido às (i) arestas bem definidas, representando as bordas de objetos, e às (ii) regiões homogêneas, representado interior de objetos e fundo da cena. É importante enfatizar que, utilizando a representação dos mapas de

profundidade em uma escala de tons de cinza, apenas informações de luminância são necessárias, o que reduz consideravelmente a quantidade de informação quando comparado aos dados de textura que necessitam também de informações de crominância.

O formato de dados MVD é uma interessante alternativa para o formato de vídeos multi-vista, já que os mapas de profundidade tendem a ser mais simples de codificar do que as vistas de textura. Utilizando técnicas leves de síntese (ou renderização) de vistas, tal como *Depth Image Based Rendering* (DIBR) (KAUFF, 2007), os mapas de profundidade podem ser aplicados, conjuntamente com as informações de textura, para gerar quantas vistas sintetizadas (também conhecidas como vistas virtuais) forem necessárias no lado do decodificador/receptor. Como apresentado na Figura 2, somente um subconjunto das vistas (textura e mapas de profundidade) são codificadas/transmitidas e o decodificador/receptor é capaz de gerar as vistas virtuais sem a necessidade de codificar/transmitir vistas intermediárias. É importante enfatizar que, apesar dos mapas de profundidade não serem exibidos para os visualizadores, eles são de extrema importância para geração das vistas sintetizadas e, assim, suas informações devem ser preservadas durante a codificação.

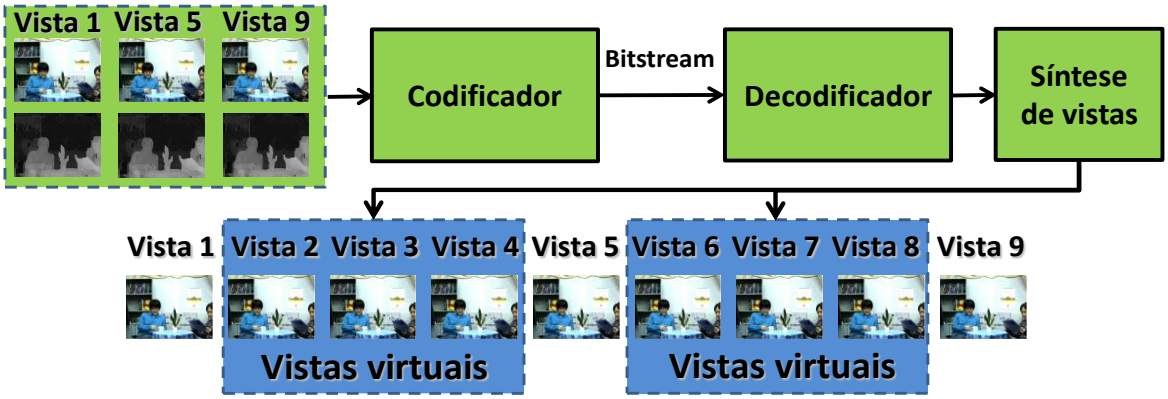


Figura 2 – Exemplo do processo de codificação e decodificação utilizando o formato de dados MVD.

1.1 Motivação

Vídeos sem codificação necessitam de quantidades proibitivas de memória para armazenamento e largura de banda para transmissão, especialmente quando vídeos de alta resolução são processados em tempo real. Este cenário é ainda pior quando são considerados vídeos 3D, que são compostos por diferentes pontos de vista.

O 3D-HEVC é uma extensão, para codificação de vídeos 3D, do padrão estado-da-arte de codificação de vídeos 2D, conhecido como HEVC. O 3D-HEVC estende as funcionalidades do HEVC, pois o padrão 2D já é capaz de alcançar elevadas taxas de compressão, mantendo elevada qualidade de visualização dos vídeos. No entanto, novas ferramentas foram inseridas e/ou modificadas/melhoradas para suportar a codificação de mais de um vídeo simultaneamente e dar suporte para a codificação dos mapas de profundidade, elevando expressivamente o custo computacional do processo de codificação. Baseado neste fato, existe um vasto espaço para exploração de soluções que possam trazer melhorias de eficiência, tanto na codificação das diferentes vistas de textura e de mapas de profundidade que representam informações semelhantes da mesma cena, quanto nas novas ferramentas de codificação especializadas nos mapas de profundidade.

Como mencionado anteriormente, os dados dos mapas de profundidade permitem a síntese de vistas, proporcionando uma redução considerável na quantidade de dados necessária para codificação e transmissão de um vídeo 3D. No entanto, durante a codificação, os mapas de profundidade acabam ocasionando um impacto considerável no custo computacional de um codificador 3D-HEVC. Este impacto é justificado por algumas novas ferramentas complexas que são responsáveis por preservar as informações dos mapas de profundidade de forma eficiente, mantendo uma elevada taxa de compressão. A preservação de informações das bordas dos mapas de profundidade é uma das tarefas mais difíceis durante a codificação e é de extrema importância para fornecer elevada qualidade nas vistas sintetizadas (que são exibidas aos espectadores).

O trabalho de SALDANHA (2015) destaca que a codificação dos mapas de profundidade representa aproximadamente 50% do tempo de codificação em um codificador 3D-HEVC. Então, técnicas eficientes para redução do tempo de execução na codificação do 3D-HEVC devem ser investigadas.

1.2 Objetivo

O objetivo deste trabalho é o desenvolvimento de uma solução para redução do tempo de codificação dos mapas de profundidade no padrão 3D-HEVC. Esta solução deve ser capaz de alcançar uma elevada redução no tempo de codificação, com um baixo impacto na qualidade das vistas que vão ser geradas a partir dos

mapas de profundidade e exibidas aos espectadores, ou seja, as vistas sintetizadas.

Para obter resultados expressivos o foco da solução abrange tanto a predição intra-quadro quanto a predição inter-quadros no codificador. Somado a isso, técnicas de mineração de dados (em inglês *Data Mining* – DM) são exploradas conjuntamente com algoritmos de aprendizado de máquina para extrair correlações entre as variáveis de codificação e reduzir a complexidade dos processos de tomada de decisão do codificador.

1.3 Organização do trabalho

Esta dissertação está dividida em sete capítulos, apresentando os principais conceitos necessários para o entendimento deste trabalho, os detalhes do desenvolvimento e os resultados alcançados. O Capítulo 2 apresenta as principais estruturas e ferramentas de codificação utilizadas pelo 3D-HEVC. Primeiramente são apresentadas as ferramentas que foram herdadas do HEVC e também são utilizadas no processo de codificação do 3D-HEVC. Após, são apresentadas as novas funcionalidades e as modificações na estrutura de codificação para o codificador 3D-HEVC. O Capítulo 3 discute os principais trabalhos relacionados encontrados na literatura. O Capítulo 4 apresenta uma análise do tempo de codificação e do comportamento das estruturas de codificação do 3D-HEVC. O Capítulo 5 detalha o desenvolvimento da solução para redução do tempo de execução na codificação dos mapas de profundidade, principal contribuição desta dissertação. No Capítulo 6 é apresentada a metodologia para as avaliações, os resultados obtidos e comparações com trabalhos relacionados. Por fim, o Capítulo 7 apresenta as conclusões e os trabalhos futuros sobre o assunto abordado.

2 3D-*HIGH EFFICIENCY VIDEO CODING*

Este capítulo tem como objetivo apresentar o fluxo, a estrutura de particionamento e o processo de codificação utilizado pelo codificador 3D-HEVC. Como o 3D-HEVC estende as funcionalidades do padrão HEVC, também se faz necessário realizar uma discussão dos passos utilizados para a codificação de vídeos no codificador 2D. Baseado neste fato, este capítulo apresenta conceitos iniciais sobre o codificador HEVC (Seções 2.1 e 2.2) que também são utilizados para o codificador 3D, além dos novos conceitos introduzidos para o 3D-HEVC.

O padrão HEVC, estado-da-arte para codificação de vídeos 2D, foi desenvolvido pelo *Joint Collaborative Team on Video Coding* (JCT-VC) (JCT-VC, 2013) e finalizado em 2013. O HEVC foi desenvolvido com o objetivo de superar a eficiência de codificação do padrão antecessor H.264/*Advanced Video Coding* (H.264/AVC) (ITU-T, 2005) e suportar a codificação de vídeos em altas definições.

O HEVC é capaz de reduzir a quantidade de informação necessária para a representação do vídeo codificado em aproximadamente 50% (SULLIVAN, 2012), quando comparado ao H.264/AVC. Esta eficiência na compressão é ocasionada pela inserção de novos modos de operação, contudo acaba gerando um elevado custo computacional no codificador.

A Figura 3 apresenta um diagrama de blocos para o codificador HEVC. Inicialmente, cada quadro do vídeo é dividido em blocos de tamanhos iguais, como será detalhado na Seção 2.1. Para o primeiro quadro, apenas a predição intra-quadro é utilizada, já que não existe nenhum quadro codificado anteriormente para servir como referência, então apenas redundâncias dentro do próprio quadro são exploradas. Os demais quadros podem utilizar tanto a predição intra-quadro quanto a predição inter-quadros, conhecidos como quadros P e B. No entanto, podem existir mais quadros I ao decorrer do vídeo, que são inseridos em determinados intervalos de tempo (mais detalhes sobre os tipos de quadros serão discutidos na Seção 2.2).

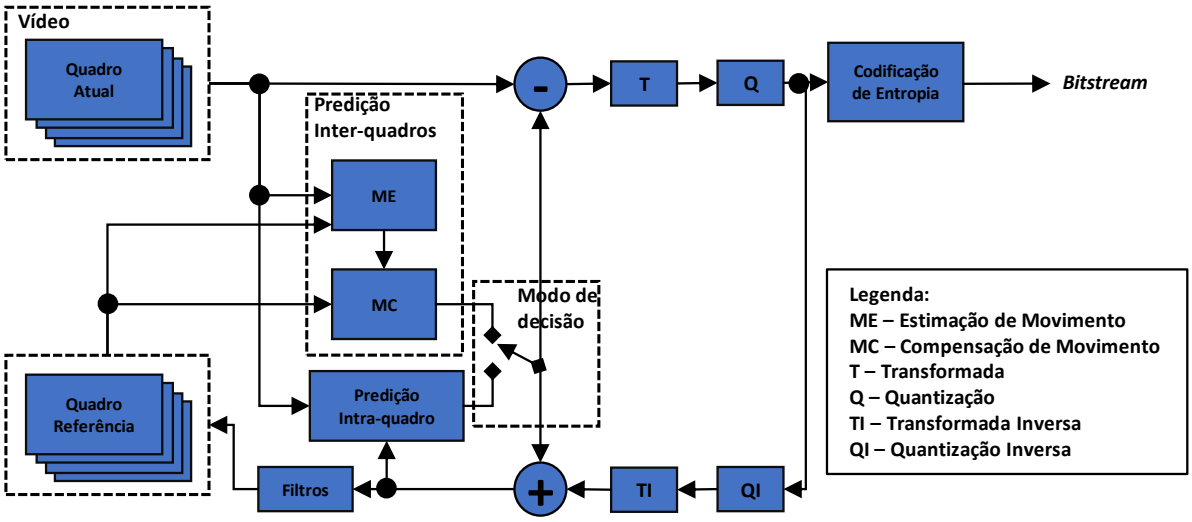


Figura 3 – Diagrama de blocos do codificador HEVC.

A predição intra-quadro utiliza apenas informações dentro do quadro atual (quadro que está sendo codificado), mais especificamente do bloco atual que está sendo codificado e dos blocos vizinhos ao bloco atual (LAINEMA, 2012). A predição inter-quadros utiliza informações do bloco do quadro atual e informações do(s) quadro(s) de referência (processados anteriormente) para localizar, nos quadros de referência, qual bloco é mais semelhante ao bloco atual (SULLIVAN, 2012).

Um modo de decisão é responsável por decidir, entre as predições intra-quadro e inter-quadros, qual predição alcança a melhor eficiência para codificar um determinado bloco.

Após computar a predição, o codificador calcula o resíduo, onde é realizada uma diferença entre o bloco do quadro atual com o bloco predito. Este resíduo passa por uma transformada espacial linear (T), quantização (Q) e codificação de entropia para gerar as informações do vídeo codificado, chamado de *bitstream*.

Como a quantização é um processo com perdas, o codificador também contém partes do processo de decodificação, para garantir que as amostras de referência sejam idênticas no codificador e no decodificador, ou seja, considerem as perdas geradas pela quantização. Portanto, com o objetivo de reconstruir a informação residual, é aplicada a quantização inversa (QI) e a transformada inversa (TI). Essa nova informação é adicionada às amostras preditas e enviada para os filtros do laço de reconstrução para compor o quadro de referência (SULLIVAN, 2012).

2.1 Estrutura de particionamento

Como mencionado anteriormente, o HEVC define que cada quadro é dividido em blocos menores. A estrutura básica de divisão de blocos no HEVC é denominada de *Coding Tree Unit* (CTU). O tamanho das CTUs é definido antes da codificação e é fixo para todas CTUs do vídeo, sem possibilidade de alteração dinâmica. Além disso, o tamanho máximo e mínimo de cada CTU está limitado a 64×64 e 8×8 amostras, respectivamente (SULLIVAN, 2012).

Cada CTU é dividida em Unidades de Codificação (UCs), em inglês *Coding Units*, em forma de uma estrutura de árvore quarternária, frequentemente mencionada pelo termo em inglês *quadtree*. Esta divisão ocorre no particionamento da CTU em quatro blocos menores de tamanhos iguais, onde novamente cada UC pode ser dividida recursivamente em quatro blocos menores. Esse processo de divisão pode ser realizado até que o tamanho mínimo de 8×8 amostras seja alcançando (SULLIVAN, 2012).

A Figura 4 exemplifica uma CTU de tamanho 64×64 (primeiro nível de profundidade) com diferentes níveis de profundidade da *quadtree*, onde as folhas da árvore (blocos azuis) são as UCs finais codificadas e o menor nível (8×8) ocorre no quarto nível de profundidade. Esta flexibilidade de particionamento permite que regiões mais detalhadas possam ser divididas em blocos menores para codificação, enquanto blocos maiores são selecionados para codificação de regiões homogêneas ou com poucos detalhes.

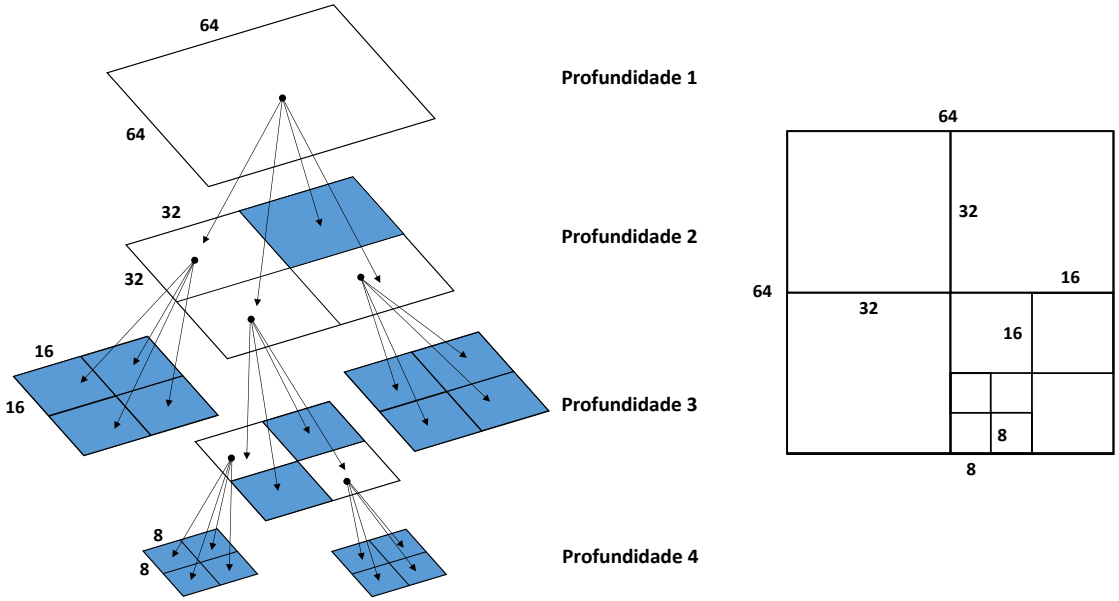


Figura 4 – Estrutura de *quadtree* de uma CTU dividida em UCs.

Além disso, cada UC pode ser dividida em uma, duas ou quatro *Prediction Units* (PUs), que são separadamente preditas. As PUs são responsáveis por armazenar informações relevantes à predição, como os vetores de movimento, o modo de codificação, particionamento da PU, entre outros. As PUs podem ser particionadas em tamanhos simétricos e assimétricos.

A Figura 5 apresenta todas representações possíveis de PUs simétricas e assimétricas (JCT-VC, 2013) para as predições intra-quadro e inter-quadros. O modo de particionamento em que uma UC é codificada como uma única PU é representado pelo particionamento 2*N*×2*N*, podendo assumir os tamanhos 64×64, 32×32, 16×16 e 8×8. Se uma UC é particionada em quatro PUs, as PUs resultantes são representadas por blocos quadráticos com o mesmo tamanho e são representados pelo particionamento *N*×*N*. Este modo de particionamento é suportado apenas para o menor tamanho de UC permitido (8×8), já que este particionamento é o equivalente a dividir um bloco em quatro UCs e codificar cada uma dessas UCs como uma única PU. Além disso, para particionar uma UC em duas PUs são permitidos seis modos de particionamento que podem ser utilizados pela predição inter-quadros.

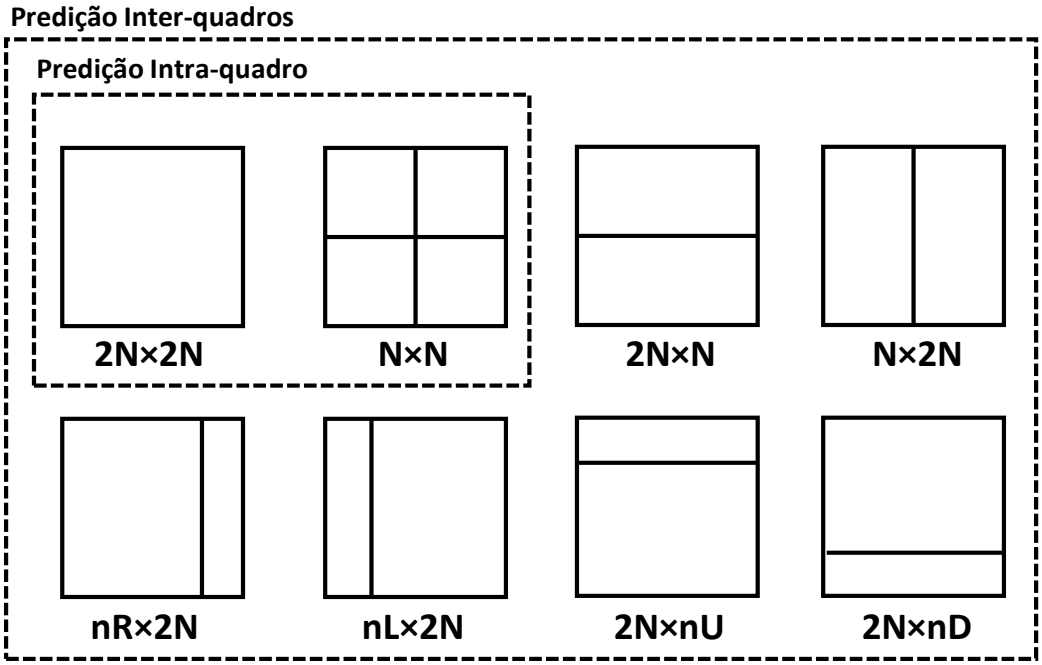


Figura 5 – Representação dos particionamentos possíveis de PU para as predições intra-quadro e inter-quadros.

2.2 Processo de predição

Em um codificador HEVC cada sequência de vídeo pode ser dividida em grupos de quadros, conhecido como *Group of Pictures* (GOP). Um GOP é uma sequência de quadros codificados contendo todas informações necessárias dentro do próprio GOP para realizar a decodificação, ou seja, sem a necessidade de informações de quadros que não pertencem ao GOP codificado. Todos os GOPs possuem o mesmo tamanho, que é definido de forma estática através de um parâmetro de configuração do codificador.

Um GOP pode conter três tipos de quadros: (i) quadro I; (ii) quadro P e (iii) quadro B. Onde quadros do tipo I são codificados utilizando apenas a predição intra-quadro. Quadros do tipo P são de predição unidirecional, onde tanto a predição intra-quadro quanto a predição inter-quadros podem ser utilizadas, mas neste caso, apenas quadros vizinhos passados são utilizados como referência na predição inter-quadros. Quadros do tipo B são de predição bidirecional; assim como os quadros P, os do tipo B podem utilizar as predições intra-quadro e inter-quadros. Porém, além dos quadros temporalmente passados, neste caso podem-se utilizar como referência quadros temporalmente futuros ao quadro atual, mas que foram codificados anteriormente (SULLIVAN, 2012).

A predição intra-quadro é responsável por remover a redundância de informações presentes em um quadro atual, utilizando apenas informações dentro do próprio quadro. Para isso, o HEVC define 35 modos de predição, sendo 33 modos direcionais, o modo DC e o modo planar (JCT-VC, 2013). Como apresenta a Figura 5, apenas partições de tamanho $N \times N$ e $2N \times 2N$ são permitidas para a predição intra-quadro, que tipicamente variam em PUs de 4×4 até 64×64 amostras.

Como a predição intra-quadro utiliza apenas informações do próprio quadro, quadros do tipo I são utilizados no primeiro quadro de um vídeo, pois não existem quadros de referência, e os demais quadros podem ser do tipo P e B. Os quadros I também podem ser utilizados em determinados intervalos de tempo, para reduzir a propagação de erros transmitida entre as referências utilizadas e melhorar a qualidade visual do vídeo. Em sistemas de transmissão contínua (*streaming*), quadros do tipo I podem ser utilizados para manter o fluxo de exibição quando referências são perdidas durante uma transmissão. Além disso, quadros I são necessários para permitir a sincronização dos vídeos em um *streaming*, quando o

usuário começa a assistir ao vídeo em um ponto aleatório da sequência e, assim, não recebeu quadros de referência anteriores.

A predição inter-quadros, utilizada em quadros do tipo P e B, é responsável por encontrar informações redundantes entre quadros temporalmente vizinhos. Comumente, os vídeos são capturados a uma elevada frequência de quadros (pelo menos 24 quadros por segundo) para permitir aos espectadores uma sensação de movimento contínuo, aumentando a quantidade de informações redundantes entre quadros temporalmente vizinhos. Com isto, a predição inter-quadros é responsável pela maior parte dos ganhos de compressão obtidos nos padrões atuais de codificação de vídeos (CHENG, 2009). No entanto, a predição inter-quadros também é a principal responsável pelo elevado tempo de processamento nos codificadores de vídeo atuais.

Para decidir o melhor modo de codificação e alcançar uma elevada eficiência de codificação (considerando taxa de bits e distorção do vídeo), o codificador HEVC adota o modo de decisão conhecido como *Rate-Distortion Optimization* (RDO) (SULLIVAN, 1998). O processo do RDO avalia as diversas possibilidades dos modos de codificação e seleciona o modo que alcançar o menor *Rate-Distortion cost* (*RD-cost*) (SULLIVAN, 2012).

2.2.1 Predição intra-quadro

A Figura 6 apresenta os 33 modos direcionais da predição intra-quadro permitidos pelo padrão HEVC. As setas indicam a primeira amostra que será utilizada para gerar a PU predita. Todas as amostras da PU sendo predita serão copiadas de amostras posicionadas no mesmo ângulo de inclinação. As amostras de referência são utilizadas das bordas dos blocos de referência à esquerda e acima do bloco que está sendo codificado.

Além dos modos direcionais, existem o modo planar e o modo DC. No modo planar, é realizada uma interpolação entre as amostras de referência da borda superior e da borda esquerda da PU para determinar as amostras preditas. No modo DC, a PU é predita com um valor médio calculado pela média de todas amostras de referência da borda superior e da borda à esquerda (LAINEMA, 2012).

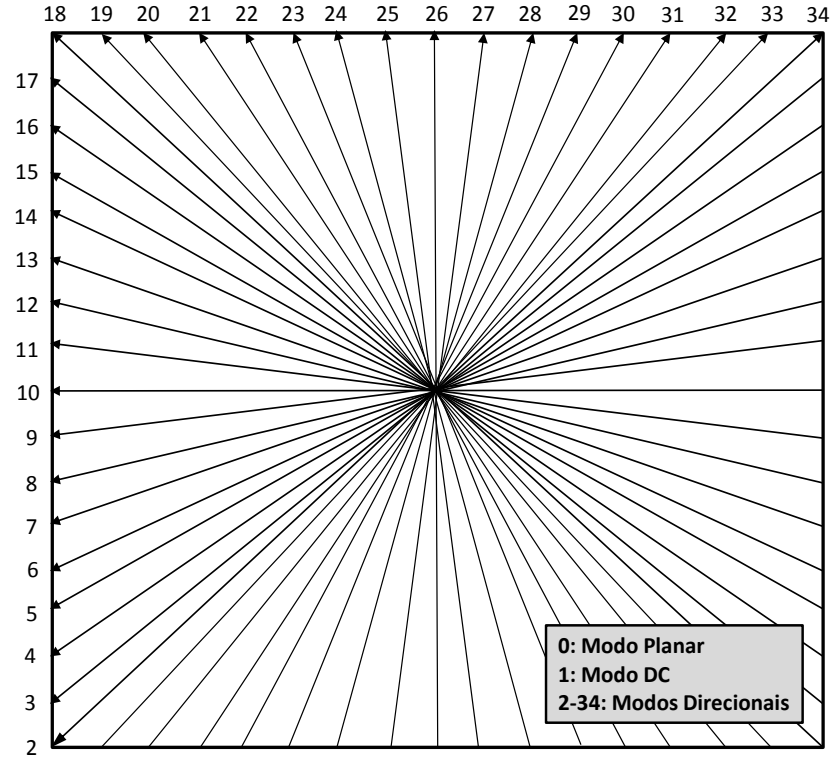


Figura 6 – Modos de predição intra-quadro no HEVC (LAINEMA, 2012).

Como a avaliação de todos os 35 modos de predição intra-quadro através do RDO é muito complexa para o codificador, o software de referência do HEVC (ROSEWARNE, 2015) utiliza um modo de decisão rápido, reduzindo os modos de predição que precisam ser avaliados pelo RDO completo.

O *Rough Mode Decision* (RMD) (ISO/IEC-JCT1/SC29/WG11, 2010) é aplicado para construir uma lista de candidatos limitada para ser enviada ao RDO, com três melhores modos para PUs 64×64, 32×32 e 16×16 e oito no caso de PUs 8×8 e 4×4. Para tanto, é realizada uma avaliação local de todas possibilidades de modos de predição utilizando o *Sum of Absolute Transformed Differences* (SATD). Além disso, também é utilizada a técnica *Most Probable Modes* (MPM), onde os modos mais prováveis são inseridos na lista de candidatos analisados pelo RDO. Estes modos mais prováveis são inferidos de PUs vizinhas previamente codificadas (LAINEMA, 2012).

A inserção do RMD e do MPM reduzem consideravelmente o tempo computacional da predição intra-quadro.

2.2.2 Predição inter-quadros

A predição inter-quadros tem como objetivo reduzir a redundância temporal entre imagens temporalmente vizinhas em uma cena. Como apresentado na Figura

3, a predição inter-quadros é dividida em duas etapas: (i) Estimação de movimento, em inglês *Motion Estimation* (ME) e (ii) Compensação de movimento, em inglês *Motion Compensation* (MC).

A ME tem como objetivo mapear o deslocamento de um bloco em relação a um quadro de referência. Para indicar esse deslocamento, a ME utiliza vetores de movimento, que são vetores em duas dimensões (*x*,*y*). Para encontrar o melhor vetor de movimento é realizada uma busca no quadro de referência para encontrar o bloco com maior semelhança ao bloco que está sendo codificado. Esta busca é limitada a uma determinada área e o centro desta área tipicamente encontra-se na posição do bloco co-localizado, como apresentado na Figura 7. A semelhança dos blocos é avaliada por um critério de similaridade, onde o *Sum of Absolute Differences* (SAD) é frequentemente utilizado. A busca é realizada por algoritmos desenvolvidos para encontrar o melhor casamento (semelhança) de blocos e com pouco tempo de processamento.

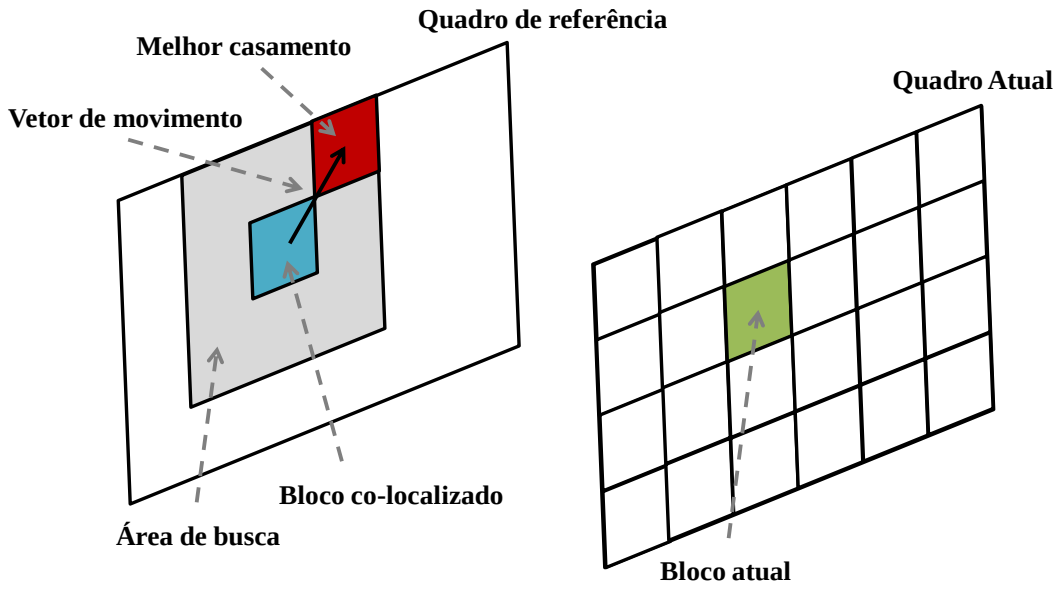


Figura 7 – Funcionamento da ME no HEVC.

Um aspecto importante da ME é que a busca pelo melhor casamento de bloco não se limita a apenas a um quadro de referência. A ME pode utilizar múltiplos quadros de referência e mantém duas listas de quadros reconstruídos para serem utilizados como referência para os próximos quadros (SULLIVAN, 2012). Uma lista é responsável por armazenar o índice dos quadros temporalmente passados na ordem de visualização. Além disso, existe outra lista que é responsável por armazenar o índice dos quadros temporalmente futuros na ordem de visualização, mas que podem ter sido codificados anteriormente. Como mencionado na Seção 2.2, os

quadros do tipo P (unidirecionais) utilizam apenas a lista dos quadros temporalmente passados, enquanto os quadros do tipo B (bidirecionais) podem utilizar informações das listas dos quadros temporalmente passados e futuros.

Por fim, a MC recebe os vetores de movimentos gerado pela ME e, com base nesses vetores, é responsável por montar o quadro predito. Este quadro será subtraído do quadro atual para geração dos resíduos e será transmitido para as etapas de transformada e quantização.

Além disso, a predição inter-quadros do HEVC permite o uso do modo merge/SKIP (*Merge*/*SKIP mode* – MSM), onde o codificador pode derivar informações de movimento de blocos espacialmente ou temporalmente vizinhos (SULLIVAN, 2012). Este modo utiliza uma lista *merge* contendo os blocos vizinhos e o índice da lista serve como referência no processo de decodificação. O modo *SKIP* é considerado um caso especial do MSM, onde não é realizada a transmissão de resíduos. Neste caso só são transmitidos para o decodificador a *flag* indicando que o modo *SKIP* foi utilizado e o índice da lista *merge*, reduzindo a quantidade de informações para representação e transmissão do vídeo.

É importante mencionar que a etapa da ME realiza um refinamento utilizando a ferramenta *Fractional Motion Estimation* (FME). Após encontrar o melhor candidato na etapa da ME considerando amostras inteiras, novas amostras fracionárias são geradas através de interpolação pela FME. Por fim, a FME busca pela melhor semelhança considerando as amostras fracionárias ao redor da melhor posição encontrada pela ME aplicada em amostras inteiras. (SULLIVAN, 2012).

As próximas subseções apresentam as novas funcionalidades inseridas para a codificação de vídeos 3D no 3D-HEVC.

2.3 Estrutura de codificação

Como já mencionado, o 3D-HEVC adota o formato de dados *Multi-View plus Depth* (MVD) (MERKLE, 2007), o qual é uma alternativa promissora para o formato de vídeos multi-vistas. No modelo MVD cada quadro de textura está associado a um mapa de profundidade correspondente. Os mapas de profundidade são comumente capturados por sensores infravermelhos que estão acoplados as câmeras responsáveis por capturar os dados de textura. Estes quadros, com informações de profundidade, são responsáveis por fornecer informações geométricas da cena,

como a distância da câmera até os objetos.

A estrutura de codificação do 3D-HEVC é baseada em *Access Units* (AUs), onde cada AU contém todos os quadros de textura e seus respectivos mapas de profundidade em um determinado instante de tempo. Para codificar cada quadro em uma AU, o 3D-HEVC utiliza uma estrutura avançada de particionamento baseado em *quadtrees*, tanto para os dados de textura quanto para os dados dos mapas de profundidade, herdada do HEVC (apresentada anteriormente na Seção 2.1).

A Figura 8 ilustra a estrutura básica de codificação do 3D-HEVC e destaca as dependências entre AUs, vistas, componentes (textura/mapas de profundidade) e os tipos de quadros codificados (quadros do tipo I, P e B).

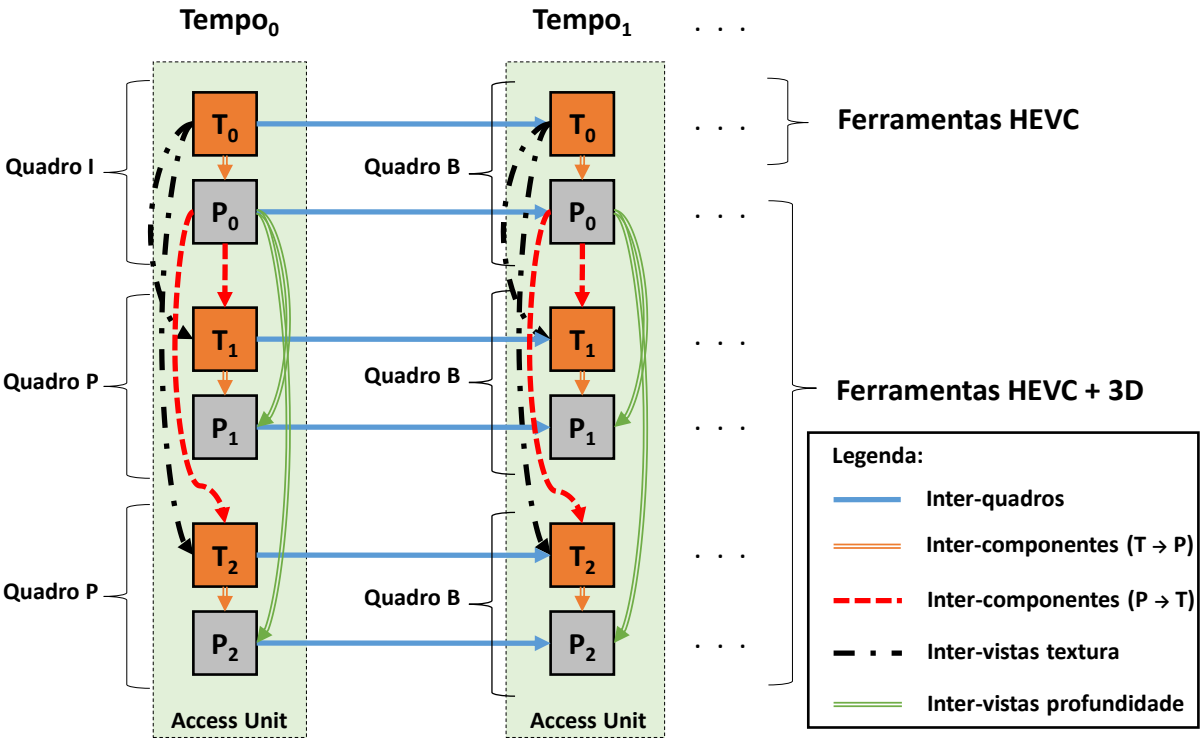


Figura 8 – Estrutura básica de codificação do 3D-HEVC com inter-quadros, inter-vistas e inter-componentes.

Uma AU pode conter um número arbitrário de vistas em que a primeira a ser codificada é frequentemente chamada de vista base (ou vista independente). É importante destacar que a Figura 8 é baseada em ferramentas do software de referência, que implementa todas funcionalidades definidas para o 3D-HEVC, conhecido como 3D-HEVC *Test Model* (3D-HTM) 16.0 (CHEN, 2015).

A Figura 8 ilustra a vista base (T_0) e duas vistas dependentes de texturas (T_1, T_2), além dos seus mapas de profundidade correspondentes (P_0, P_1, P_2). A vista T_0 é independente das outras vistas e é codificada com o HEVC, já que um

decodificador do HEVC deve ser capaz de fornecer imagens para um dispositivo de exibição 2D convencional, a partir de um *bitstream* gerado pelo codificador 3D-HEVC (CHEN, 2015). Contudo, as vistas dependentes podem ser codificadas utilizando informações da vista base, de forma a reduzir as redundâncias presentes entre as vistas, e também utilizar informações dos mapas de profundidade. Esta codificação, explorando a redundância entre as vistas, é realizada pela ferramenta *Disparity-Compensated Prediction* (DCP) (GUO, 2005).

Além disso, a Figura 8 demonstra que o 3D-HEVC utiliza uma estrutura hierárquica baseada em quadros B, onde o primeiro quadro da vista base e seu mapa de profundidade correspondente são quadros do tipo I, e os remanescentes são quadros do tipo B. Enquanto os primeiros quadros das vistas dependentes são do tipo P e os remanescentes são do tipo B.

Um codificador HEVC, sem nenhuma modificação, é responsável por codificar os dados de textura de quadros do tipo I no 3D-HEVC e, neste caso, somente a predição intra-quadro do HEVC (apresentada na Subseção 2.2.1) está disponível para realizar a predição. Contudo, quando os mapas de profundidade de quadros I são codificados, o fluxo de codificação avalia os modos da predição intra-quadro do HEVC e as novas ferramentas para predição intra-quadro do 3D-HEVC, tal como *Depth Modeling Modes* (DMM) (MERKLE, 2015), *Segment-wise Direct Component Coding* (SDC) (LIU, 2014) e *Depth Intra Skip* (DIS) (LEE, 2015).

Em adição às ferramentas utilizadas para quadros do tipo I, para quadros de textura e mapas de profundidade em quadros do tipo B da vista base, a estimação de movimento bidirecional também é explorada para reduzir a redundância temporal (detalhes na Seção 2.2.2), aumentando a probabilidade de alcançar uma maior taxa de compressão, mas ao custo de um aumento significativo no custo computacional.

Um codificador 3D-HEVC aplica as seguintes ferramentas para codificar quadros das vistas dependentes (quadros do tipo P): (i) predição intra-quadro do 3D-HEVC; (ii) predição inter-vistas com DCP para dados de textura e mapas de profundidade; (iii) ferramentas inter-componentes explorando dados de textura para codificar os mapas de profundidade; e (iv) ferramentas inter-componentes visando acelerar a codificação de textura explorando informações dos mapas de profundidade através de ferramentas, tal como *Depth Based Block Partitioning* (DBBP) (JAGER, 2013). Em quadros do tipo B, as vistas dependentes utilizam as mesmas ferramentas dos quadros P, mas com o adicional da estimação de

movimento bidirecional.

A estrutura de codificação demonstrada na Figura 8 considera o 3D-HTM sendo executado para a configuração de codificação *Random Access* (RA), onde quadros I, P e B são utilizados. No entanto, também é possível realizar a execução com a configuração *All-Intra* (AI) e, neste caso, apenas quadros I são considerados.

Além disso, o 3D-HEVC também possibilita codificar apenas a informação de textura (i.e., sem mapas de profundidade associados) e os mesmos algoritmos de codificação estão disponíveis, apenas removendo os algoritmos específicos para codificação dos mapas de profundidade. No entanto, sempre que os mapas de profundidade estiverem presentes, eles devem estar necessariamente associados às vistas de textura.

É importante ressaltar que um codificador 3D deve ser capaz de fornecer dados codificados em um *bitstream* que possam ser decodificados por um decodificador 2D. Neste caso, o 3D-HEVC não insere dependências de mapas de profundidade, nem de outras vistas na codificação da vista base, para que ela possa ser decodificada por um decodificador 2D convencional (CHEN, 2015).

Quando os mapas de profundidade estão associados aos dados de textura na codificação é definido um par de valores para o parâmetro de quantização (*Parameter Quantization* – QP), como por exemplo QP=(25, 34), representando que a vista base de textura deve ser codificada utilizando o valor de QP 25 e a vista base do mapa de profundidade com o valor de QP 34. O valor do QP indica a intensidade das perdas inseridas na etapa da codificação dos resíduos e está diretamente relacionado a eficiência de codificação, que é considerada como uma relação entre a taxa de bits e a distorção do vídeo. Quanto maior for o valor do QP, maiores serão os cortes aplicados nos resíduos e maior será a taxa de compressão, mas por outro lado tende a proporcionar maior perda de informação e ocasiona maior perda na qualidade do vídeo.

Todas as ferramentas citadas anteriormente para a codificação dos quadros I, P e B para os dados de textura e dos mapas de profundidade serão detalhadas a seguir.

2.4 Codificação de textura

A codificação da vista independente é feita por um codificador HEVC sem

nenhuma modificação. Por outro lado, a codificação das vistas dependentes utiliza os mesmos conceitos presentes no padrão HEVC, porém são adicionadas novas ferramentas que consideram informações já utilizadas em uma vista previamente codificada, para possibilitar uma representação mais eficiente das vistas dependentes com uma maior exploração de redundância dos dados.

Como o foco deste trabalho não está relacionado a codificação de textura, as próximas subseções apresentam apenas uma breve visão geral das principais ferramentas inseridas na codificação que são utilizadas no 3D-HEVC.

2.4.1 Predição compensada pela disparidade – DCP

Como uma primeira ferramenta introduzida para a codificação das vistas dependentes, em adição à predição compensada pelo movimento (do inglês *Motion-Compensated Prediction* – MCP), composta pela ME e a MC, o 3D-HEVC utiliza o conceito de estimação de disparidade (*Disparity-Compensated Prediction* – DCP) (GUO, 2005). No entanto, padrões de codificação de vídeos 3D anteriores, como o H.264/*Multi-view Video Coding* (MVC) (VETRO, 2011), já utilizavam a DCP. Devido ao grande potencial de ganhos na codificação, o 3D-HEVC optou por incorporar este conceito no desenvolvimento do codificador.

Como apresentado na Figura 9, a MCP (destacada em azul e verde) refere-se a uma predição a quadros que já foram codificados na mesma vista, ou seja, é aplicada entre quadros temporalmente vizinhos, enquanto a DCP (destacada em vermelho) é aplicada entre quadros de vistas vizinhas, mas em um mesmo instante de tempo (na mesma AU).

Assim como a MCP, a DCP divide o quadro que está sendo codificado em vários blocos e cada bloco é buscado em uma área ao redor do bloco co-localizado no quadro de uma vista previamente codificada. Para realização da busca, é necessário aplicar um algoritmo com o objetivo de encontrar o bloco mais semelhante possível dentro da área selecionada. Após encontrar o melhor bloco disponível, é gerado um vetor de disparidade apontando para esta posição e o resíduo entre o bloco que está sendo codificado e o bloco encontrado como melhor candidato para predição é enviado para as próximas etapas de codificador.

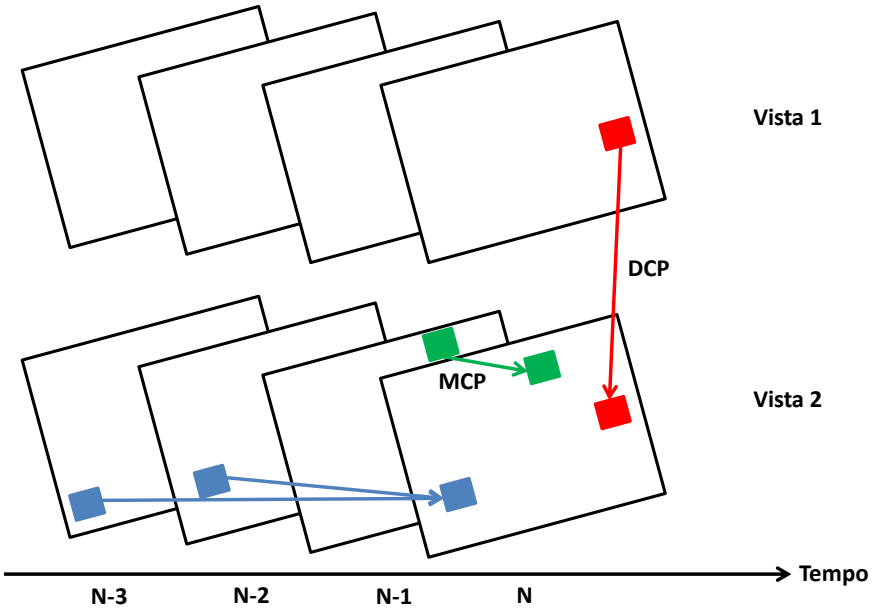


Figura 9 – Exemplo de DCP e MCP, adaptado de (CHEN, 2015).

2.4.2 Predição de movimento inter-vistas – IMP

A ferramenta de predição de movimento inter-vistas (*Inter-view Motion Prediction* – IMP) assume valores estimados das amostras do mapa de profundidade para a imagem de textura que está sendo codificada. Para derivação dos parâmetros de movimento para um bloco atual em uma vista dependente, o maior valor de uma amostra do bloco associado do mapa de profundidade é convertido a um vetor de disparidade. Adicionando o vetor de disparidade à localização da amostra central do bloco atual que está sendo codificado resulta em uma localização de uma amostra dentro de um bloco em um quadro já codificado de uma vista de referência. Este bloco é utilizado como bloco de referência.

Se o bloco de referência foi codificado utilizando a MCP, os parâmetros de movimento associados podem ser usados como parâmetros de movimento candidatos para o bloco atual na vista que está sendo codificada. O vetor de disparidade também pode ser utilizado diretamente como um candidato para a DCP (CHEN, 2015).

2.4.3 Predição de resíduos avançada – ARP

A ferramenta *Advanced Residual Prediction* (ARP) explora a correlação entre os resíduos obtidos de diferentes vistas ou diferentes AUs. Esta ferramenta é baseada no fato de que as informações de movimento e disparidade de diferentes vistas ou de quadros temporalmente vizinhos possuem uma forte correlação. Deste

modo, ARP realiza a predição de resíduos de uma PU 2N×2N considerando o resíduo obtido no bloco correspondente da vista de referência ou em um quadro previamente codificado.

Somente a diferença entre o resíduo do bloco atual e o predito é enviada nos dados codificados do *bitstream* (LI, 2013).

2.4.4 Compensação de iluminação – IC

A compensação de iluminação (*Illumination Compensation* – IC) consiste em um modelo linear utilizado para adaptar a iluminação dos blocos preditos pelo processo de predição inter-vistas, para que fique compatível com a iluminação da vista atual. Os parâmetros do modelo linear são estimados para UC utilizando amostras reconstruídas do bloco atual e do bloco de referência utilizado para predição (CHEN, 2015).

2.4.5 Particionamento de bloco baseado nos mapas de profundidade – DBBP

No modelo de particionamento de bloco baseado no mapa de profundidade (*Depth-Based Block Partitioning* – DBBP), o particionamento do bloco de textura é realizado e derivado a partir de uma máscara de segmentação binária, computada a partir do mapa de profundidade correspondente em uma vista de referência.

Como apresenta a Figura 10, para geração da máscara de segmentação binária, primeiramente, é identificado o bloco do mapa de profundidade na vista de referência. Este bloco tem o mesmo tamanho do bloco atual de textura. Após, é calculado um limiar baseado na média de todas as amostras do mapa de profundidade dentro do bloco de referência. Então, a máscara de segmentação é gerada baseada nos valores do mapa de profundidade e no limiar encontrado. As regiões do particionamento são definidas pela regra: caso o valor da amostra do mapa de profundidade seja maior que o limiar, a máscara binária assume o valor um, caso contrário assume o valor zero. Por fim, a compensação de movimento é realizada em cada uma das partições geradas (CHEN, 2015).

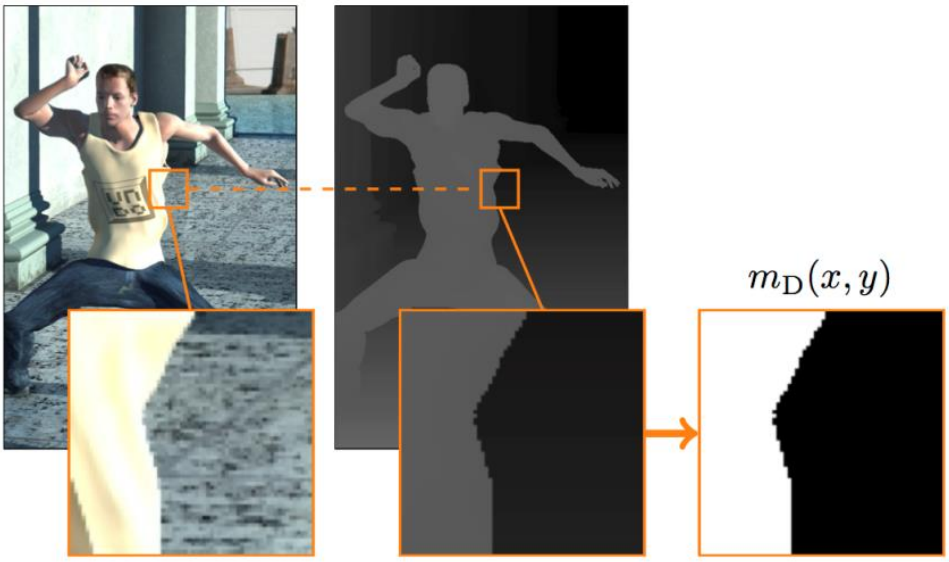


Figura 10 – Ilustração da geração de máscara binária a partir de um bloco de profundidade para a sequência de vídeo *Undo_Dancer* (CHEN, 2015).

2.5 Codificação de mapas de profundidade

Vídeos 3D que não utilizam informações de profundidade associadas aos dados de textura necessitam estimar a profundidade de objetos presentes na cena, possibilitando a inserção de erros durante a codificação. Visando contornar este problema e reduzir a quantidade de informações para representação de vídeos 3D, o 3D-HEVC adota o modelo de representação de profundidade em informações contidas nos mapas de profundidade, que são associados a quadros de textura. Como mencionado anteriormente, este modelo é conhecido como MVD e permite a síntese de vistas intermediárias com maior qualidade, quando comparado a sistemas que necessitam inferir a profundidade dos objetos a partir de vistas de textura.

Alguns conceitos presentes na codificação de textura também são utilizados para a codificação de mapas de profundidade. Contudo, algumas ferramentas foram alteradas, outras ferramentas foram desabilitadas e algumas ferramentas foram adicionadas para a codificação de mapas de profundidade. Esta seção tem como objetivo apresentar as principais modificações no codificador para dar suporte à codificação de mapas de profundidade.

2.5.1 Eliminação do filtro interpolador

Uma das tarefas mais difíceis durante a codificação dos mapas de profundidade é a preservação de arestas durante a codificação. O filtro interpolador

utilizado na MCP, mais especificamente pela FME, tende a degradar as informações das arestas em mapas de profundidade. Essas perdas de informações nas arestas dos mapas de profundidade podem produzir erros nas vistas sintetizadas, que são visíveis aos espectadores.

A desabilitação do filtro interpolador na codificação dos mapas de profundidade elimina o problema de distorções nas arestas, além de possibilitar uma redução no custo computacional do codificador e do decodificador. Esta restrição também é estendida para a DCP onde apenas a estimação de disparidade com precisão de pixel inteiro está habilitada (CHEN, 2015).

2.5.2 Eliminação da filtragem *in-loop*

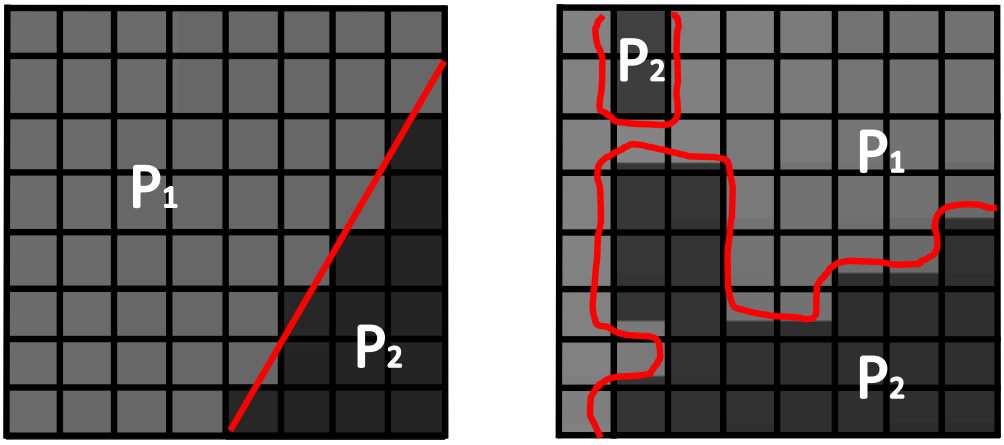
Os filtros *in-loop* do HEVC foram desenvolvidos para codificação de imagens de textura. Esses filtros foram inseridos no laço de decodificação para suavizar e reduzir as distorções inseridas durante a codificação realizada em blocos (CHEN, 2015). Por outro lado, estes filtros tendem a ocasionar perda de qualidade nas bordas dos mapas de profundidade, reduzindo a qualidade das vistas sintetizadas. Portanto, os filtros (i) *Deblocking Filter* (DF) (NORKIN, 2012) e (ii) *Sample Adaptive Offset* (SAO) (FU, 2012) foram desabilitados na codificação dos mapas de profundidade.

2.5.3 *Depth Modeling Modes* - DMM

Os mapas de profundidade apresentam duas características importantes: arestas bem definidas e grandes áreas homogêneas. Os modos de predição intra-quadro do HEVC (apresentados na Subseção 2.2.1) são mantidos na predição intra-quadro dos mapas de profundidade no 3D-HEVC, pois eles alcançam eficiência elevada quando codificam regiões homogêneas ou com poucas variações entre os valores das amostras. Contudo, estes modos de predição podem gerar artefatos nas vistas sintetizadas quando codificam regiões com arestas. Visando contornar este problema, dois Modos de Modelagem de Profundidade (*Depth Modeling Modes* - DMM) são adicionados na predição intra-quadro dos mapas de profundidade no 3D-HEVC: (i) DMM-1 e (ii) DMM-4.

Os DMMs aproximam os blocos de mapa de profundidade por um modelo que particiona o bloco em duas regiões e define um valor constante para cada região, conhecida como valor constante da partição (*Constant Partition Value* – CPV)

(CHEN, 2015). Dois tipos de partições podem ser utilizados: (i) *Wedgelet* e (ii) Contorno. Partições *wedgelet* são separadas em duas regiões, representadas por uma linha reta, que separa as regiões P_1 e P_2, como apresenta a Figura 11(a). Essa separação é determinada por um padrão de *wedgelet* que define um ponto inicial e um ponto final da reta, ambos localizados nas bordas de um bloco. No domínio discreto, a equação da reta passa pelo meio de alguns blocos, permitindo verificar se o pixel está mais presente em uma região ou em outra. Com isso, cada amostra é mapeada para um valor binário, que informa se ela pertence à região P_1 ou P_2.



(a) (b)

Figura 11 – Predição de blocos dos mapas de profundidade por (a) wedgelet e (b) contorno.

Um exemplo de partição por contorno é apresentado na Figura 11(b), onde as regiões podem ser desconexas e ter um formato totalmente arbitrário, como é o caso de P_2. Como os contornos não possuem um padrão para verificação de casamento (semelhança), a verificação de qual região cada pixel pertence é feita através da comparação de cada pixel com um limiar, que será detalhado posteriormente.

As Subseções 2.5.3.1 e 2.5.3.2 apresentam maiores detalhes do DMM-1 e do DMM-4, que são utilizados na predição intra-quadro dos mapas de profundidade.

2.5.3.1 Sinalização explícita de wedgelet usada no DMM-1

O princípio básico do DMM-1 é encontrar o melhor casamento (semelhança) de partição *wedgelet* durante a codificação e transmitir a informação da partição para o *bitstream*. Durante a decodificação, o sinal do bloco é reconstruído utilizando a informação da partição *wedgelet* transmitida.

Para a codificação, o modo DMM-1 é dividido em três etapas principais: (i) inicialização da lista de padrões *wedgelet* para cada tamanho de bloco; (ii) busca em

uma lista de padrões *wedgelet* pelo padrão com a menor distorção e (iii) refinamento.

A inicialização da lista de padrões de *wedgelets* é pré-definida para cada bloco de 32×32 até 4×4 amostras. Após, é feita uma busca na lista do conjunto de possíveis partições *wedgelets* para cada tamanho de bloco disponível. Esta busca é realizada utilizando, como referência, o sinal original do bloco de mapa de profundidade que está sendo codificado. A busca é efetuada com o objetivo de encontrar a partição *wedgelet* com a menor distorção entre o sinal do bloco original e a aproximação realizada pela partição *wedgelet* selecionada. Por fim, a *wedgelet* que obtiver o melhor resultado é enviada para o processo de refinamento e posições adjacentes ao padrão encontrado são avaliadas, para verificar se um resultado melhor é alcançado (CHEN, 2015).

2.5.3.2 Predição inter-componente de partições de contorno usada no DMM-4

A ideia principal do DMM-4 é realizar a predição de um bloco de mapa de profundidade a partir de uma partição de contornos utilizando o bloco co-localizado de textura como referência. Portanto, a vista de textura é utilizada como referência, fornecendo informações do sinal de luminância reconstruído do bloco de textura co-localizado.

A predição por contornos é realizada por uma estratégia que utiliza um limiar para comparar as amostras do bloco do mapa de profundidade e permite a divisão deste em regiões distintas. Este limiar é calculado como o valor médio dos quatros cantos do sinal de luminância do bloco co-localizado de textura utilizado como referência. O particionamento é realizado comparando cada pixel do bloco do mapa de profundidade com o limiar encontrado. Caso a amostra do mapa de profundidade comparada com o limiar seja maior que a média, esta amostra é atribuída à uma região P_1, caso contrário a amostra é atribuída à uma região P_2 (CHEN, 2015).

2.5.4 *Segment-wise Direct Component Coding* – SDC

A abordagem da ferramenta *Segment-wise Direct Component Coding* (SDC) fornece uma alternativa aos métodos convencionais de codificação residual para obter ganhos de compressão em regiões homogêneas. O SDC explora a ideia de que as regiões dos mapas de profundidade tendem a possuir poucas variações entre as amostras de um bloco (ou uma PU).

Com a utilização do SDC, os dados residuais são codificados sem passar

pelos processos de transformada e quantização. Ao invés de utilizar a informação residual das amostras do mapa de profundidade, o SDC considera que um único resíduo chamado de *Direct Component* (DC) codifica a região predita pelos modos intra-quadro do HEVC e dois resíduos codificam as regiões preditas pelos DMMs.

Para gerar o DC é necessário computar o DC predito e o DC original, e então realizar a subtração dos valores. O DC original é obtido realizando o cálculo da média das amostras do bloco original (sem realizar a predição) para a predição intra-quadro do HEVC e o cálculo da média das amostras de cada região para os DMMs. O DC predito para a região que realizou a predição com os modos da predição intra-quadro do HEVC é gerado utilizando o valor da média das amostras dos quatros cantos do bloco predito, enquanto que o DC predito para os DMMs utiliza o *Constant Partition Value* (CPV), obtido aplicando os procedimentos detalhados em (ZHAO, 2013).

É importante enfatizar que o tamanho da partição de uma UC contendo uma PU codificada utilizando o SDC é sempre 2N×2N. O SDC pode ser aplicado em todos os modos de codificação da predição intra-quadro dos mapas de profundidade, o que inclui os modos intra-quadro do HEVC (apresentados na Subseção 2.2.1) e os modos DMMs, e também na predição inter-quadros (também codifica apenas um valor DC).

2.5.5 *Depth Intra Skip* – DIS

A ferramenta de predição *Depth Intra Skip* (DIS) leva em consideração que os mapas de profundidade são compostos, em grande parte, por regiões homogêneas, que são representadas por valores iguais ou muito próximos. As amostras dos mapas de profundidade presentes em uma região homogênea não têm grande impacto na qualidade da síntese de vistas e, frequentemente, compartilham o mesmo valor de representação. Com isso, a ferramenta DIS tem como objetivo realizar uma codificação com baixo custo computacional, sem codificar e transmitir resíduos, reduzindo a quantidade de informações transmitidas para o *bitstream*. Portanto, esta ferramenta envia para o *bitstream* apenas a informação de qual modo de predição do DIS foi utilizado para o bloco do mapa de profundidade (CHEN, 2015).

Para efetuar a predição dos blocos de mapa de profundidade, a ferramenta DIS utiliza quatro modos (dois herdados da predição intra do HEVC) que estão

representados na Figura 12. O modo vertical (a) é o mesmo utilizado na predição intra-quadro do HEVC (replicando em toda coluna as amostras de referência, do bloco acima já codificado), enquanto o modo vertical (b) utiliza somente a amostra de referência central para predizer todo o bloco. Os modos horizontais (c) e (d) utilizam o mesmo conceito dos modos verticais, mas utilizam como referência as amostras do bloco à esquerda, previamente codificado.

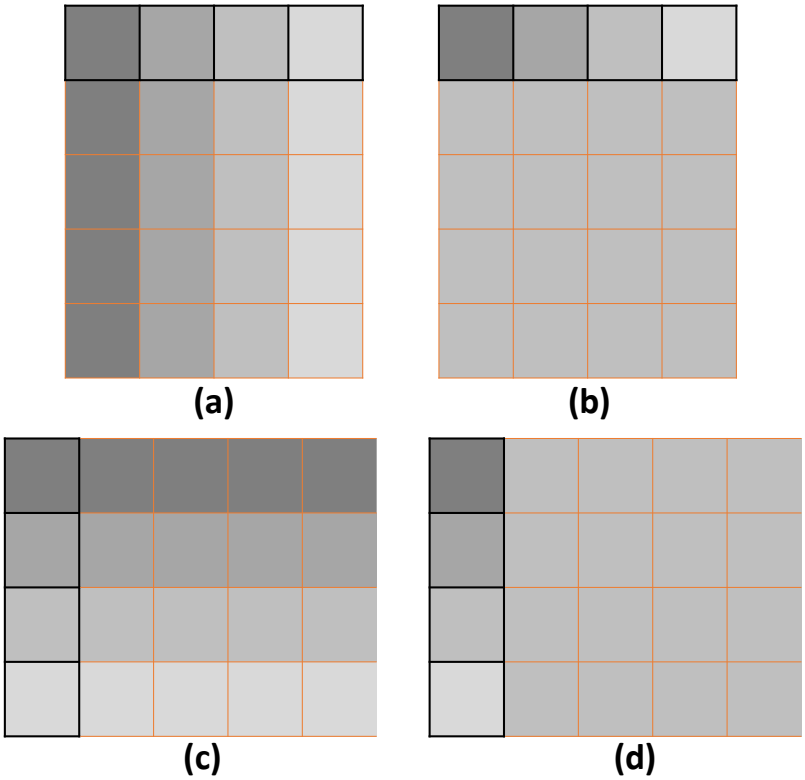


Figura 12 – Modos de predição para a ferramenta DIS: (a) e (b) modos verticais e (c) e (d) modos horizontais.

2.5.6 Predição intra-quadro dos mapas de profundidade

Apresentado o funcionamento dos novos modos da predição intra-quadro dos mapas de profundidade no 3D-HEVC, esta subseção tem como objetivo detalhar o fluxo de codificação e como esses modos atuam junto da predição intra-quadro do HEVC em um codificador 3D-HEVC.

A predição intra-quadro dos mapas de profundidade no 3D-HEVC é composta de quatro principais ferramentas de predição: (i) predição intra-quadro do HEVC; (ii) DMM-1; (iii) DMM-4; e (iv) DIS, como apresentado na Figura 13. Com a definição de quais modos da predição intra do HEVC, do DMM-1 e do DMM-4 devem ser avaliados pelo RDO completo (Lista RD), estes modos são codificados pela Codificação Residual (transformadas e quantização) e também pelo SDC. A seguir,

o resultado da Codificação Residual, do SDC e do DIS são processados pelo codificador de entropia para que o *RD-cost* seja calculado.

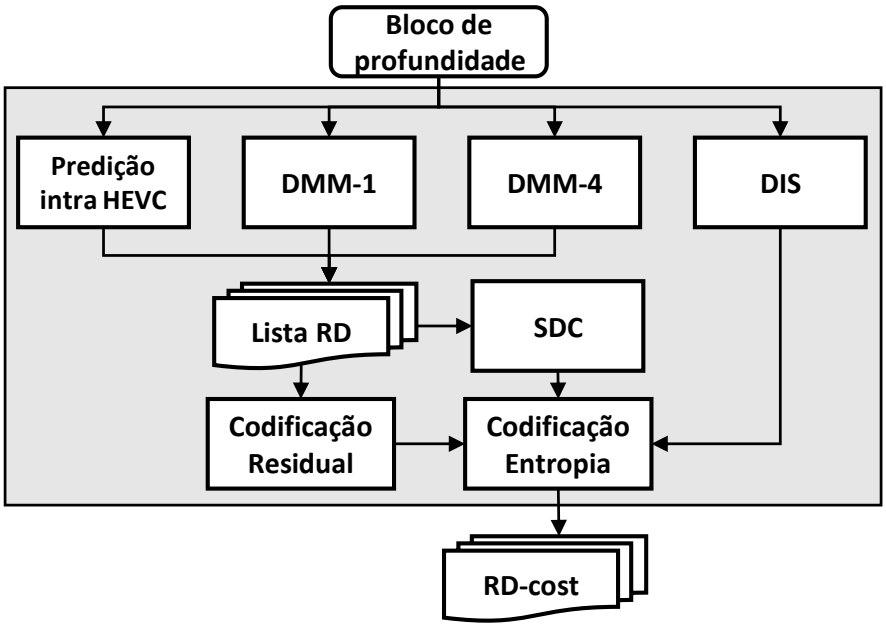


Figura 13 – Diagrama de blocos da predição intra-quadro dos mapas de profundidade no 3D-HEVC.

As ferramentas de predição intra-quadro do HEVC e DIS são selecionadas, frequentemente, quando uma região no bloco do mapa de profundidade compartilha o mesmo valor ou valores similares, enquanto os DMMs tendem a ser escolhidos quando regiões compostas por arestas estão sendo codificadas.

2.5.7 Herança dos parâmetros de movimento – MPI

A ferramenta herança dos parâmetros de movimento (*Motion Parameter Inheritance* – MPI) trabalha baseada na ideia de que as características de movimento do sinal do vídeo (textura) e seu mapa de profundidade associado devem ser semelhantes, visto que ambos são projeções da mesma cena, do mesmo ponto de vista e no mesmo instante de tempo.

Para aumentar a eficiência da codificação dos dados dos mapas de profundidade, os parâmetros de movimento do bloco de textura podem ser herdados para codificar o bloco do mapa de profundidade. Portanto, em adição às informações já contidas na lista *merge*, como informações de PUs espacialmente e temporalmente vizinhas, também são adicionadas informações do bloco de textura previamente codificado (CHEN, 2015). O codificador seleciona qual PU da lista *merge* será utilizada como referência e copia as informações desta PU para a PU que está sendo codificada.

Como os quadros de textura podem utilizar precisão de um quarto de pixel e

os mapas de profundidade possuem somente precisão inteira, o processo de herança de parâmetros de movimento pode ser quantizado para a posição mais próxima com precisão inteira (CHEN, 2015).

2.5.8 Predição das *quadtrees* dos mapas de profundidade – QTL/QTPred

A predição das *quadtrees* dos mapas de profundidade é composta por duas ferramentas: *Quadtree Limitation* (QTL) (MORA, 2014) e *Quadtree Prediction* (QTPred) (ZHANG, 2014). QTL é responsável por inferir o limite do particionamento da *quadtree* dos mapas de profundidade, a partir da *quadtree* de textura. Assim como a ferramenta apresentada no Subseção 2.5.7, esta ferramenta é baseada na ideia de que os mapas de profundidade possuem elevada correlação com os dados de textura e, portanto, as suas *quadtrees* também estão correlacionadas. Então, a ferramenta QTL define que a *quadtree* dos mapas de profundidade não pode apresentar mais divisões do que a *quadtree* de textura. Além disso, a ferramenta QTPred restringe o particionamento da PU do mapa de profundidade: os particionamentos verticais (N×2N, nR×2N, nL×2N) ou horizontais (2N×N, 2N×nU, 2N×nD) só são permitidos em uma UC de mapa de profundidade quando a textura for codificada com partições verticais/horizontais ou N×N.

É importante destacar que as ferramentas QTL e QTPred não são utilizadas para quadros do tipo I (quadros codificados somente com predição intra-quadro).

A Figura 14 apresenta os possíveis particionamentos de uma PU para dados textura e os particionamentos permitidos para os mapas de profundidade, quando a QTPred é utilizada.

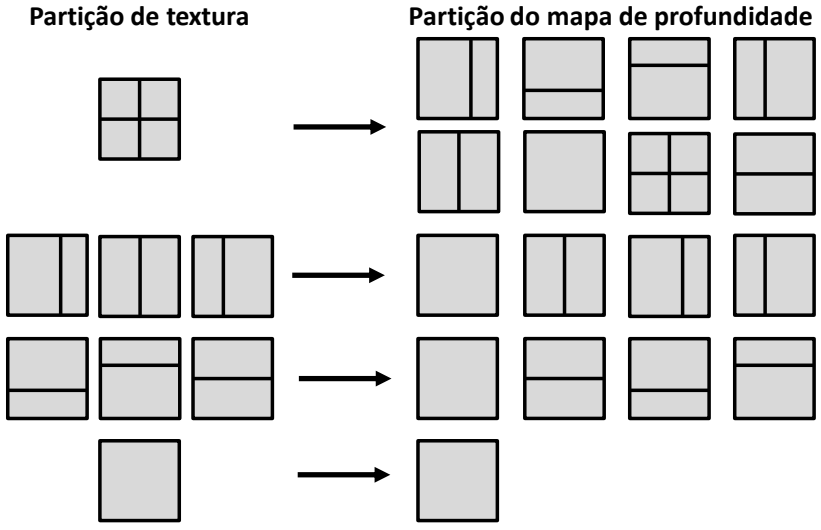


Figura 14 – Possíveis partições para UCs de textura e correspondentes partições permitidas para UCs de mapas de profundidade (CHEN, 2015).

2.5.9 *View Synthesis Optimization* – VSO

A informação geométrica fornecida pelos mapas de profundidade não é visível, sendo explorada somente no processo de síntese (ou renderização) de vistas, de forma que erros na codificação dos mapas de profundidade causam perdas na qualidade das vistas sintetizadas. Baseado neste fato, o cálculo da distorção obtido no modo de decisão dos mapas de profundidade é modificado de forma a levar em consideração tanto a distorção da vista sintetizada quanto a distorção do mapa de profundidade. Para obter a distorção das vistas sintetizadas, duas métricas são utilizadas no processo do RDO.

A primeira métrica é o *Synthesized View Distortion Change* (SVDC) (TECH, 2012), que para ser computada, necessita de funcionalidades de renderização na codificação. Como o custo computacional, em termos de tempo de processamento, é um fator crítico para o cálculo da distorção, um renderizador com menor complexidade e com funcionalidades mais básicas é inserido no codificador para verificar o impacto da codificação do mapa de profundidade na qualidade das vistas sintetizadas.

A segunda métrica é a aplicação do SVDC sem renderização, ou seja, um modelo baseado na estimação da distorção das vistas sintetizadas sem renderização. A ideia básica desta métrica é estimar a distorção da vista sintetizada pela ponderação da distorção do mapa de profundidade com um fator derivado a partir da vista de textura. Esta métrica tem como objetivo reduzir o custo computacional de utilizar o renderizador no processo de codificação como ocorre na métrica anterior (MA, 2014).

Com as ferramentas apresentadas neste capítulo foi possível verificar que existem diversas possibilidades de modos de particionamento e modos de codificação que são avaliadas durante a codificação gerando um elevado custo computacional no codificador 3D-HEVC. Para reduzir esse problema diversas técnicas foram desenvolvidas visando reduzir esse elevado custo computacional e essas técnicas serão apresentadas no próximo capítulo.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados à codificação de vídeos 3D tendo por base o codificador 3D-HEVC e que são de interesse para essa dissertação. Uma busca extensiva na literatura foi realizada e trabalhos com diferentes abordagens de redução de tempo de codificação foram encontrados. Diversos trabalhos visam a aceleração da codificação dos dados de textura com a utilização de técnicas inter-componentes, heurísticas para as etapas de estimação e disparidade de movimento, entre outros. No entanto, como este trabalho foca na codificação dos mapas de profundidade, este capítulo restringe-se nas soluções que foram consideradas mais relevantes no escopo de redução de tempo na codificação dos mapas de profundidade.

3.1 Redução do tempo na codificação dos mapas de profundidade do 3D-HEVC

Existem diversos trabalhos na literatura propondo técnicas que visam reduzir o tempo da codificação nos mapas de profundidade no codificador 3D-HEVC. Nos trabalhos (PENG, 2016), (ZHANG, 2016a), (MORA, 2014), (LEI, 2016), (ZHANG, 2016b), (FU, 2015), (JABALLAH, 2016), (MA, 2015), (KIM, 2015) e (SANCHEZ, 2014b) são utilizadas diferentes estratégias que focam em diferentes etapas para acelerar a codificação dos mapas de profundidade. Estes trabalhos são capazes de fornecer diferentes pontos de operação para um codificador 3D-HEVC, quando consideradas a redução de tempo de execução e a eficiência de codificação.

No trabalho de (ZHANG, 2016b) foi proposta uma solução para redução do tempo da predição intra-quadro dos mapas de profundidade. A solução é baseada em um modo de decisão com objetivo de avaliar apenas os modos mais promissores para serem utilizados e, assim, simplificar o fluxo de codificação do software 3D-HTM. O modo de decisão é baseado na comparação de um limiar com o menor *RD-cost* dos modos intra-quadro do HEVC avaliados e tem como objetivo enviar apenas o modo com o menor *RD-cost* direto para a codificação de resíduos com o SDC, evitando o fluxo tradicional que também avalia a codificação de resíduos passando pelas transformadas e quantização. Além disso, para integrar a solução foi desenvolvida uma métrica menos complexa do que a utilizada no 3D-HTM para comparar as partições avaliadas no DMM-1.

Os trabalhos (FU, 2015) e (JABALLAH, 2016) desenvolveram heurísticas para redução do tempo de processamento na predição intra-quadro com foco no DMM-1. Como o 3D-HTM realiza uma busca exaustiva entre as possíveis partições disponíveis para avaliação, estes trabalhos utilizam informações do bloco que está sendo codificado e relacionam com as partições avaliadas com o objetivo de reduzir o número de *wedgelets* avaliadas. Também para o DMM-1, (MA, 2015) desenvolveu uma solução visando reduzir a quantidade de memória utilizada para armazenamento das *wedgelets*. A solução realiza subamostragem das informações das *wedgelets* para blocos de tamanhos 8×8 e 4×4 a partir de informações de blocos de tamanho 16×16, ou seja, apenas as informações das *wedgelets* de blocos 16×16 precisam ser armazenadas.

KIM (2015) desenvolveu uma nova ferramenta de codificação para a predição intra-quadro como alternativa aos DMMs e a predição intra do HEVC. Esta ferramenta codifica as regiões homogêneas dos mapas de profundidade com baixo impacto no tempo de execução do codificador e reduz o número de informações para representação do bloco. Após algumas modificações, esta ferramenta foi incorporada ao software de referência 3D-HTM com o nome *Depth Intra Skip* (DIS), que foi apresentada na Subseção 2.5.5.

O trabalho de (SANCHEZ, 2014b) apresenta uma técnica baseada em um detector de arestas para simplificar o fluxo de codificação da predição intra-quadro. Neste trabalho, o *Simplified Edge Detector* (SED) é responsável por classificar um bloco como aresta ou região homogênea. Baseado nesta classificação, é decidido se os modos DMM-1 e DMM-4 devem ser avaliados ou não. Caso um bloco seja classificado como aresta, os modos DMM-1 e DMM-4 são avaliados normalmente e nenhuma simplificação é realizada. Caso contrário, apenas os modos intra-quadro do HEVC são avaliados; i.e., os modos DMMs não são avaliados.

3.1.1 Redução do tempo de execução para a *quadtree* dos mapas de profundidade

Embora existam diversos trabalhos para redução de tempo de processamento dos mapas de profundidade, apenas alguns deles focam em técnicas para reduzir o tempo de computação da *quadtree*.

Em (PENG, 2016) foi proposta uma solução que abrange dois níveis (bloco e *quadtree*) da estrutura de codificação dos mapas de profundidade para quadros I.

Em nível de bloco, foi desenvolvida uma técnica baseada no *RD-cost* dos modos da predição intra-quadro. Os *RD-cost* gerados durante a avaliação dos modos intra-quadro do HEVC e os DMMs são comparados com determinados limiares. Estes limiares são gerados dinamicamente, durante a codificação de alguns quadros sem nenhuma simplificação, com o objetivo de fornecer um modo de decisão capaz de evitar o fluxo completo da predição intra-quadro dos mapas de profundidade. O algoritmo em nível de *quadtree* computa a variância da UC que está sendo codificada e a máxima variância dos sub-blocos dentro dessa UC, com o objetivo de decidir se a UC deve ser dividida em UCs menores. A divisão da UC só ocorre se a máxima variância dos sub-blocos for maior que a variância da UC ou a variância da UC é maior que um determinado limiar.

O trabalho desenvolvido em (ZHANG, 2016a) também propõe uma solução em nível de *quadtree* considerando apenas quadros I. Uma técnica baseada em *Corner Points* (CPs) (HARRIS, 1988) é utilizada para identificar regiões compostas por arestas nos mapas de profundidade que tendem a ser mais difíceis de serem codificadas e utilizam UCs menores, com objetivo de realizar uma predição do nível de profundidade da *quadtree* de uma determinada UC. A predição da profundidade da *quadtree* permite pular a avaliação de UCs maiores e também evitar a avaliação de UCs menores.

O trabalho de (MORA, 2014) explora uma solução inter-componente para acelerar a codificação dos mapas de profundidade limitando a *quadtree*, baseado nas informações da UC correlacionada de textura que foi previamente codificada. Esta técnica parte da ideia de que os dados de textura e mapas de profundidade representam a mesma cena, do mesmo ponto de vista ao mesmo instante de tempo. Então, as *quadtrees* dos dados textura e dos mapas de profundidade devem estar altamente correlacionadas. Baseado nisso, durante a codificação dos quadros P e B, uma UC de mapa de profundidade nunca é particionada em um tamanho menor do que a UC de textura correlacionada. É importante destacar que esta solução está incorporada ao software de referência 3D-HTM e é aplicada apenas para os quadros P e B.

Em (LEI, 2016) foi proposto um algoritmo de modo de decisão rápido para a codificação de mapas de profundidade baseado na similaridade da escala de tons de cinza, que representa o bloco do mapa de profundidade, e na correlação entre as vistas. Este modo de decisão é composto por uma decisão antecipada para as UCs

e um modo de decisão para as PUs das vistas dependentes. Na decisão antecipada para UCs, um limiar é responsável por classificar a escala de tons de cinza entre a UC que está sendo codificada e o bloco co-localizado no quadro de referência. Quando existir uma elevada similaridade entre os blocos, o nível da profundidade da *quadtree* da UC que está sendo codificada é limitado ao nível da *quadtree* do bloco co-localizado no quadro de referência. O modo de decisão para as PUs das vistas dependentes utiliza a similaridade da escala de tons de cinza e as correlações entre as vistas. Quando existir uma elevada similaridade entre as PUs das vistas dependentes com a correlacionada da vista independente e determinado modo de codificação for escolhido, as PUs das vistas dependentes realizam a avaliação apenas deste modo, evitando a avaliação dos demais modos de codificação.

3.2 Considerações sobre os trabalhos relacionados

Analisando os trabalhos relacionados é possível notar que as soluções propostas para redução do tempo na codificação da *quadtree* dos mapas de profundidade exploram poucas características da UC que está sendo codificada e não exploram técnicas que possam correlacionar essas características eficientemente. Analisando os trabalhos propostos por (PENG, 2016) e (LEI, 2016), por exemplo, as soluções consideram apenas as características das amostras dentro da UC de profundidade que está sendo codificada para definir (com base em limiares) quando deve ser realizado o particionamento da UC em tamanhos menores. Estes trabalhos são capazes de alcançar bons resultados de redução no tempo de processamento (apresentados no Capítulo 6). No entanto, por utilizarem técnicas que não permitem extrair e correlacionar as características mais relevantes durante a codificação essas soluções tendem a realizar algumas decisões erradas, degradando a eficiência de codificação.

Como as UCs possuem diversas características e parâmetros de codificação que influenciam nas tomadas de decisão do codificador, esse trabalho visa utilizar técnicas de aprendizado de máquina para desenvolver uma solução que identifique e correlacione as características e parâmetros mais relevantes durante a codificação de UCs dos mapas de profundidade, que possam ser utilizados para a tomada de decisão para a divisão das UCs de forma mais eficiente.

4 AVALIAÇÕES SOBRE A CODIFICAÇÃO DO 3D-HEVC

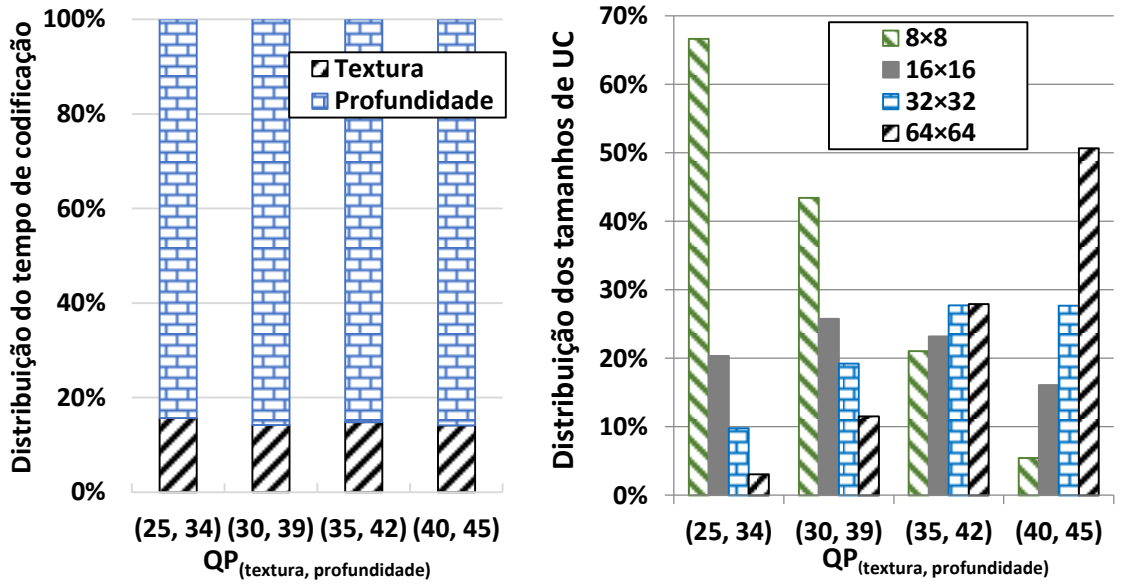
Este capítulo apresenta um conjunto de avaliações sobre o codificador 3D-HEVC, visando compreender o comportamento dos componentes (textura e mapas de profundidade) e a estrutura de particionamento dos mapas de profundidade. Através destas avaliações foi possível estabelecer as prioridades e as estratégias para a solução proposta nesta dissertação.

Para estas avaliações inicias, foram consideradas as configurações *All-Intra* (AI) e *Random Access* (RA) do codificador 3D-HEVC, conforme explicado na Seção 2.3. A metodologia de avaliação considerou as Condições Comum de Teste (CCTs) (MULLER, 2014) definidas para o 3D-HEVC (detalhes na Seção 6.1). Assim, foram consideradas oito sequências de vídeo (podem ser visualizadas no Apêndice C) que foram avaliadas considerando quatro pares de QPs, conforme definido nas CCTs e explicado na Seção 2.3. O software de referência utilizado nas avaliações foi o 3D-HTM 16.0 (CHEN, 2015).

Para cada experimento, foi avaliado o tempo gasto na codificação dos dados de textura e dos mapas de profundidade, fornecido pelo software de referência 3D-HTM. Além disso, foram implementadas funcionalidades adicionais no 3D-HTM com o objetivo de obter informações sobre a distribuição dos tamanhos de UCs. Todos os dados foram obtidos utilizando as configurações e as sequências das CCTs para o 3D-HEVC. No entanto, é importante enfatizar que como este trabalho propõe uma solução para redução do tempo de execução para a codificação das *quadtrees* dos mapas de profundidade, as análises foram realizadas com a ferramenta QTL/QTPred (Subseção 2.5.8) desativada, e no capítulo que apresenta os resultados da solução proposta nessa dissertação, é realizada a comparação com esta ferramenta.

A Figura 15 (a) apresenta a distribuição de tempo de execução entre textura e mapas de profundidade para diferentes pares de QP, considerando a configuração AI. Nesta figura é possível observar que o codificador gasta mais de 80% do tempo de codificação para os dados de mapas de profundidade e menos de 20% para os dados de textura. Isto ocorre no cenário de configuração AI, pois a codificação de textura aplica somente a predição intra-quadro do HEVC, enquanto que a codificação dos mapas de profundidade utiliza também as ferramentas DMM-1, DMM-4, DIS e SDC, além da predição intra-quadro do HEVC. Neste cenário, a

codificação dos mapas de profundidade utiliza 5,8 vezes mais tempo de execução do que a codificação de textura, em média.



(a) (b)

Figura 15 – Distribuição para configuração de codificação AI: (a) tempo de processamento para codificação de textura e mapas de profundidade; e (b) tamanhos de UCs para codificação dos mapas de profundidade.

A Figura 15 (b) apresenta a utilização dos diferentes tamanhos de UC na codificação dos mapas de profundidade para diferentes pares de QP, também considerando a configuração AI. É possível perceber na Figura 15 (b) que a distribuição das escolhas dos tamanhos de UCs não é uniformemente distribuída. É possível notar que as variações nos valores dos pares de QPs causam diferentes distribuições nos tamanhos de UCs.

Valores de QPs maiores utilizados durante a codificação conduzem a uma maior perda de informação do vídeo, gerando mais regiões homogêneas e essas regiões são codificadas de forma eficiente com tamanhos de UCs maiores, tal como 64×64. Por outro lado, valores de QPs menores tendem a preservar mais as áreas heterogêneas do vídeo e áreas com mais detalhes tendem a ser codificadas com tamanhos de UCs menores para alcançar uma boa eficiência de codificação. Por exemplo, para o QP=(25, 34) aproximadamente 67% das UCs foram codificadas com tamanho 8×8 e somente 3% das UCs foram codificadas com tamanho 64×64. Em contraposição, para o QP=(40, 45), que é o maior valor de QP avaliado, aproximadamente 50% das UCs foram codificadas com tamanho 64×64, e somente 5% foram codificadas com tamanho 8×8.

A Figura 16 (a) apresenta a distribuição do tempo gasto para a codificação dos dados de textura e de mapas de profundidade para o codificador na configuração RA. Para a configuração RA, a codificação de textura utiliza um percentual de tempo de execução superior ao encontrado na configuração AI. O caso mais equilibrado ocorre para os valores de QP=(25, 34), onde a codificação de textura ocupa aproximadamente 31% do tempo gasto no codificador. Na média, esta análise demonstrou que a codificação dos mapas de profundidade é três vezes mais custosa, em termos de tempo de execução, do que a codificação de textura.

A Figura 16 (b) apresenta o uso dos diferentes tamanhos de UC na codificação dos mapas de profundidade, também considerando a configuração RA. Os resultados na Figura 16 (b) demonstram que para QPs mais baixos existe uma distribuição mais equilibrada entre os tamanhos de UCs utilizados, quando comparado ao comportamento do AI. Por outro lado, a elevação do valor de QP causa um crescimento acelerado no uso de tamanhos de UCs maiores. Com QP=(40, 45), por exemplo, mais do que 90% dos tamanhos de UCs são escolhidos como 64×64. Por outro lado, para valores de QPs menores (QP=(25, 34)), que requerem um vídeo com maior qualidade, apenas aproximadamente 30% das UCs são codificadas com tamanho 64×64.

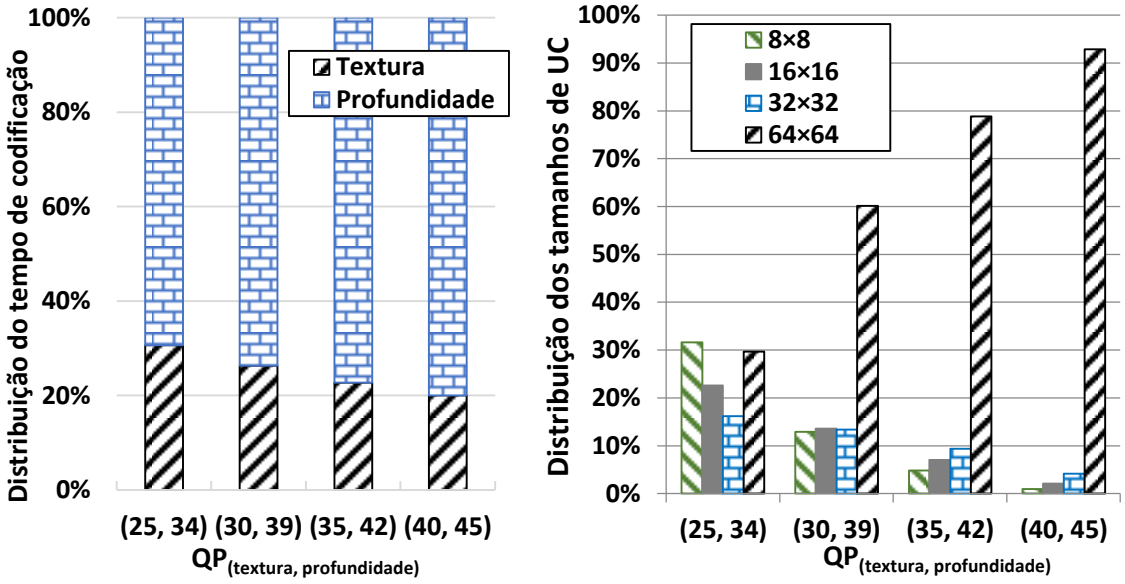


Figura 16 – Distribuição para a configuração de codificação RA: (a) tempo de execução para codificação de textura e mapas de profundidade; e (b) tamanhos de UC para codificação dos mapas de profundidade.

A Figura 17 apresenta a distribuição do tempo de codificação gasto para codificar cada tamanho de UC dos mapas de profundidade, considerando cada par de QP e utilizando a configuração AI como exemplo. Analisando esta figura é possível notar que existe um alto custo computacional ao avaliar UCs de tamanho 8×8, alcançando mais de 40% do tempo de processamento quando considerado o QP=(40, 45). Enquanto que, as UCs 64×64 ocupam a menor fatia do tempo de codificação ao serem avaliadas, para todos os casos, indicando que uma decisão antecipada nos primeiros níveis de profundidade da *quadtree* pode evitar o alto custo de processamento do codificador 3D-HEVC.

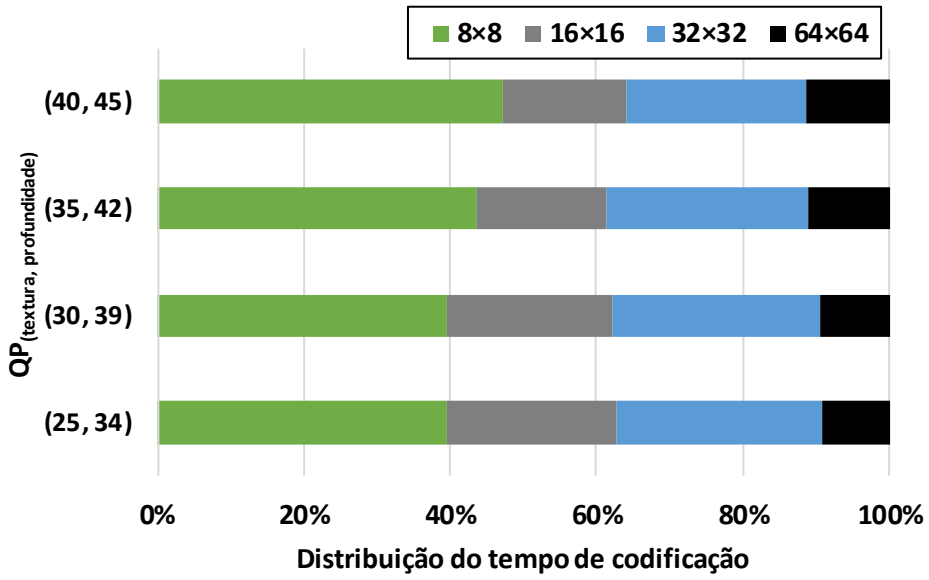


Figura 17 – Distribuição do tempo de codificação para cada tamanho de UC dos mapas de profundidade na configuração AI.

4.1 Considerações sobre as análises do codificador

Com base nas análises apresentadas, é possível inferir que uma solução rápida capaz de decidir quando uma UC deve ser dividida em UCs menores pode evitar o elevado custo de avaliar o processo completo do RDO (avaliação de diversos modos de codificação decidindo pelo modo de menor custo de taxa e distorção). Esta solução pode considerar características dos mapas de profundidade e dos parâmetros do codificador (detalhes no Capítulo 5), tal como o valor do QP.

Além disso, uma solução especializada para quadros I (que tendem usar UCs menores para valores de QPs baixos e UCs maiores para valores de QPs altos) e uma solução especializada para quadros P e B (que tendem usar em muito dos casos UCs de tamanho 64×64), pode reduzir o tempo de codificação, mantendo a

elevada eficiência de codificação.

O codificador realiza muitas decisões em tempo de execução que determinam a eficiência da codificação. Consequentemente, soluções que possam definir estaticamente algumas dessas decisões, considerando o contexto do codificador (e não o processo completo do RDO) e com baixo impacto na eficiência de codificação, são altamente desejáveis.

Técnicas baseadas em mineração de dados, em inglês *Data Mining* (DM), podem ser usadas para identificar correlações entre variáveis dependentes, identificando regularidades e construindo generalizações nos atributos do conjunto de dados. As árvores de decisão são modelos construídos através da mineração de dados, que são frequentemente utilizadas quando a solução requer alta precisão e baixo tempo de execução (APTÉ, 1997). Essas características são essenciais para a solução desenvolvida nesta dissertação, já que esta visa alcançar uma significativa redução no tempo de processamento mantendo a eficiência de codificação.

O trabalho de (CORREA, 2015) utiliza mineração de dados para avaliar os atributos do codificador, visando extrair informações e correlações entre estes atributos para evitar algumas decisões dinâmicas no codificador, reduzindo o custo computacional da codificação de vídeos de textura no cenário 2D. A proposta desta dissertação também é utilizar mineração de dados, mas para reduzir o custo computacional de codificação dos mapas de profundidade em um codificador 3D. Além disto, a solução apresentada nesse trabalho propõe definir árvores de decisão especializadas para diferentes tipos de quadros. Sendo assim, o cenário é completamente diferente, necessitando de novos estudos sobre os dados de mapas de profundidade, novas avaliações das ferramentas de codificação e uma nova análise das correlações entre as variáveis. Estas avaliações conduziram para a definição de novos atributos e permitiram a definição de novas árvores de decisão para as *quadtrees* dos mapas de profundidade. Os estudos e a definição dos atributos, juntamente com a definição das árvores de decisão, são apresentados no Capítulo 5.

5 SOLUÇÃO PARA REDUÇÃO DO TEMPO DE EXECUÇÃO NA CODIFICAÇÃO DOS MAPAS DE PROFUNDIDADE

Este capítulo apresenta a solução desenvolvida para redução do tempo de execução na codificação de mapas de profundidade, que é dividida em duas partes que são baseadas em mineração de dados para a construção de árvores de decisão. Baseada nas análises apresentadas no Capítulo 4, a solução foi desenvolvida com árvores de decisão especializadas para codificação de quadros que utilizam somente a predição intra-quadro (quadros I) e árvores de decisão especializadas para codificação de quadros com predição unidirecional e bidirecional (quadros P e B). A metodologia realizada durante o desenvolvimento da solução é detalhada na Seção 5.1. A avaliação dos atributos do codificador e a definição das árvores de decisão são apresentadas na Subseção 5.2.1 para os quadros I e na Subseção 5.2.2 para os quadros P e B.

5.1 Metodologia

O 3D-HEVC define UCs quadráticas de tamanho N×N com N ∈ {8, 16, 32, 64}. Consequentemente, foram construídas árvores de decisão estáticas para decidir quando as UCs de tamanho 64×64, 32×32 e 16×16 devem ser divididas em UCs de tamanhos menores. Com a inserção das árvores de decisão no codificador, o fluxo tradicional do 3D-HTM com o RDO é alterado quando em um determinado nível da *quadtree* a decisão é de não realizar a divisão da UC e então finalizar sua codificação.

Durante a codificação, três árvores de decisão são responsáveis pela decisão de UCs de tamanhos 64×64, 32×32 e 16×16, pertencendo aos quadros I, e outras três pela decisão de UCs dentro de quadros P (unidirecional) e B (bidirecional). Para a definição das árvores de decisão e o processo de DM, a sequência de vídeo 3D *Kendo* (MULLER, 2014) foi codificada para cada par de valores de QPs definidos nas Condições Comuns de Testes (CCTs), considerando as configurações de codificação AI e RA. Para cada UC codificada, foram armazenadas informações consideradas relevantes para o processo de decisão e uma *flag* indicando se a UC foi dividida.

É importante enfatizar que existe um número limitado de sequências de vídeos 3D com seus mapas de profundidade disponíveis para experimentos de

codificação de vídeos 3D. Sendo assim, a sequência de vídeo *Kendo* foi selecionada aleatoriamente, a partir do conjunto de vídeos disponíveis nas CCTs e, portanto, utilizada para o processo de treinamento *off-line* (responsável pela definição das árvores de decisão). Além disso, apenas uma sequência de vídeo foi utilizada no processo de treinamento para evitar o sobreajuste nos dados de treinamento (*overtraining*). No entanto, todas as sequências de vídeos definidas nas CCTs foram avaliadas (Capítulo 6) para demonstrar que as soluções treinadas são capazes de alcançar elevada eficiência em diferentes cenários de codificação.

O *Waikato Environment for Knowledge Analysis* (WEKA) (HALL, 2009) na versão 3.8, foi utilizado para auxiliar no processo de DM descrito neste capítulo. O WEKA é de acesso livre, dispõe de ferramentas *open-source* que incluem diversos algoritmos de aprendizado de máquina e fornece as seguintes funcionalidades para tratamento de dados: (i) pré-processamento; (ii) classificação; (iii) clusterização; e (iv) visualização. O treinamento das árvores de decisão foi realizado utilizando o algoritmo *open-source REPTree* (QUINLAN, 1987) disponível no WEKA, onde as funcionalidades implementadas para lidar com os atributos são herdadas do algoritmo C4.5 (QUINLAN, 1993).

O *REPTree* também poda a árvore de decisão utilizando o *Reduced Error Pruning* (REP) (BRUNK, 1991), que inicia pelas folhas da árvore onde cada ponto de decisão (nodo) é substituído pela decisão (classe) com maior probabilidade de ser escolhida. Caso a precisão na predição não seja afetada, a mudança é mantida e o processo continua iterativamente até que a precisão na predição, ao realizar a poda, seja pior do que a solução anterior gerada. Este procedimento reduz a profundidade das árvores de decisão geradas, permitindo uma melhor generalização no conjunto de dados e evitando o problema do *overfitting*.

O problema de dados desbalanceados também foi considerado. Esse problema ocorre quando existe uma quantidade significativamente maior de instâncias pertecendo a uma classe do que as outras. Para reduzir este problema, o conjunto de dados de entrada foi organizado de forma que 50% dos dados possuem a decisão de dividir as UCs e 50% possuem a decisão de não dividir as UCs.

As acurácias de todas as árvores de decisão obtidas foram avaliadas pelo WEKA aplicando um processo de validação cruzada *10-fold*. A acurácia é avaliada considerando a porcentagem de decisões corretas pela quantidade total de instâncias utilizadas no processo de treinamento. Para verificar a eficiência das

soluções, todas as árvores de decisão obtidas no processo de treinamento foram implementadas no software 3D-HTM e avaliadas seguindo uma metodologia de testes com avaliações individual e combinada de todas as árvores de decisão (Capítulo 6).

A Tabela 1 apresenta os principais parâmetros utilizados nos experimentos para cada configuração do codificador. “GOP” refere-se ao tamanho da sequência de quadros que contém todas as informações necessárias que podem ser completamente decodificadas dentro do próprio GOP. “Período Intra” está relacionado ao intervalo de quadros em que cada quadro do tipo I será codificado. “Bit depth” indica a quantidade de bits utilizada para representar uma amostra dentro de um quadro do vídeo. “$\text{Vistas}_{(\text{textura, profundidade})}$” apresenta quantas vistas para cada componente foram utilizadas, e “$\text{QP}_{(\text{textura, profundidade})}$” é o valor do QP de cada componente. Todos esses parâmetros, relacionados à estrutura da codificação, estão em conformidade com as especificações das CCTs.

Tabela 1 – Configurações utilizadas nos experimentos.

Parâmetro	All-Intra	Random Access
GOP	1	8
Período Intra	1	24
Bit depth	8	
$\text{Vistas}_{(\text{textura, profundidade})}$	(3, 3)	
$\text{QP}_{(\text{textura, profundidade})}$	(25, 34), (30, 39), (35, 42), (40, 45)	
DMM	Habilitado	
DIS	Habilitado	
VSO	Habilitado	
QTL/QTPred	Desabilitado	
Rate control	Desabilitado	
Versão 3D-HTM	16.0	

As ferramentas de codificação DMM, DIS e VSO estão habilitadas seguindo as especificações das CCTs e foram detalhadas nas Subseções 2.5.3, 2.5.5 e 2.5.9, respectivamente. Assim como no Capítulo 4, todas avaliações realizadas nesta etapa estão com a ferramenta QTL/QTPred (detalhada na Subseção 2.5.8) desabilitada. O “*Rate Control”* permite especificar a taxa de bits que determinado vídeo pode atingir durante a codificação seguindo as recomendações das CCTs, esta ferramenta permanece desabilitada. Por fim, todas as avaliações desta

dissertação são realizadas no software de referência do 3D-HEVC, o 3D-HTM, na versão 16.0.

5.2 Avaliação dos atributos do codificador e definição das árvores de decisão

Diversos atributos foram avaliados para definir quais eram os mais relevantes para construir as árvores de decisão estáticas com o objetivo de codificar mapas de profundidade. As mesmas configurações de experimentos descritas na seção anterior (considerando às CCTs) foram utilizadas nesta investigação. Uma quantidade significativa de dados foi coletada dos mapas de profundidade da sequência de vídeo e das variáveis internas da codificação, para encontrar características que poderiam ajudar na decisão de divisão das UCs. Esta análise foi dividida em duas etapas. A Subseção 5.2.1 apresenta os atributos que foram analisados e as árvores de decisão treinadas para quadros do tipo I (somente predição intra-quadro), enquanto que a Subseção 5.2.2 demonstra o mesmo processo, mas para quadros do tipo P e B (predição unidirecional e bidirecional).

5.2.1 Avaliação dos atributos e definição das árvores de decisão para quadros intra (I)

Para o cenário de codificação de quadros I foram considerados atributos que tendem a fornecer informações relevantes sobre os mapas de profundidade. A distribuição dos valores das amostras, por exemplo, pode indicar diferentes informações conforme estes dados são avaliados. Quando a média e a variância das amostras são calculadas, por exemplo, é possível obter informações de distância da UC do mapa de profundidade (em relação a câmera) e a identificação de UCs homogêneas (ou heterogêneas), respectivamente. Além disso, variáveis internas do codificador, tal como o *RD-cost* obtido ao codificar determinada UC, pode indicar se este deve realizar mais avaliações para melhorar a eficiência de codificação ou terminar o processo.

Sendo assim, os seguintes atributos foram armazenados durante a execução do codificador 3D-HTM para cada tamanho de UC codificado:

- ***QP*** – É o valor atual do *QP* utilizado na codificação dos mapas de profundidade. O *QP* tem relação direta com a taxa de compressão e tem um impacto significativo na decisão de divisão das UCs.

- ***Custo T-D*** – É o custo de codificar a UC atual. Este atributo refere-se ao *RD-cost* e foi utilizado para avaliar a relação entre as decisões do codificador e a eficiência de codificação.
- ***Var*** – É a variância das amostras originais dentro da UC atual e indica a homogeneidade do bloco e, então, pode indicar se o bloco precisa ser dividido em tamanhos menores ou não.
- ***Var_tam*** – É a máxima variância de blocos menores dentro da UC atual e representa a máxima variância das amostras dentro de um bloco. Para uma UC 64×64 existem quatro instâncias deste atributo, um para cada partição de bloco possível (4×4, 8×8, 16×16 e 32×32). Esta informação pode ser útil para indicar a homogeneidade ou a presença de arestas em blocos menores.
- ***Média*** – É o valor médio das amostras da UC atual. Esta informação indica se a UC que está sendo codificada foi capturada perto ou longe da câmera. Detalhes de objetos próximos da câmera devem ser mantidos e, neste caso, é interessante avaliar tamanhos de UCs menores.
- ***Dif_max*** – É a máxima diferença entre as amostras da UC atual. Esta informação pode ser útil na decisão de divisão das UCs, pois pode indicar variações abruptas nos valores das amostras do bloco.
- ***Grad*** – É a máxima diferença absoluta dos quatros cantos da UC atual. Esta informação ajuda a indicar a presença de informações de arestas dentro da UC e estas informações devem ser preservadas durante a codificação.
- ***Grad_tam*** – É o maior gradiente de blocos menores dentro da UC atual e significa a máxima diferença absoluta dos quatros cantos destes blocos. Assim como o *Var_tam*, existem quatro instâncias deste atributo para as UCs 64×64 e essa informação pode indicar quando a UC deve ser dividida ou não.

Estes atributos foram selecionados com o objetivo de identificar regiões compostas por arestas nos mapas de profundidade, que são mais difíceis de codificar e, consequentemente, tendem a causar uma decisão de divisão da UC (SALDANHA, 2015), no entanto, após as avaliações nem todos foram utilizados na

solução proposta nesta dissertação, como será detalhado nesta seção.

A Figura 18 apresenta as curvas de densidade de probabilidade das UCs de tamanho 64×64, quando não são divididas em tamanhos menores, para alguns exemplos de atributos coletados. A Figura 18 (a) e a Figura 18 (b) apresentam as curvas para os atributos *Dif_max* e *Var_64* e evidenciam que nas regiões com valores mais baixos destes atributos, existe uma grande probabilidade de as UCs não serem divididas. A distribuição do *Custo T-D* é apresentada na Figura 18 (c), e também demonstra uma elevada correlação com a decisão de divisão das UCs, onde valores menores de *Custo T-D* indicam uma maior eficiência na codificação.

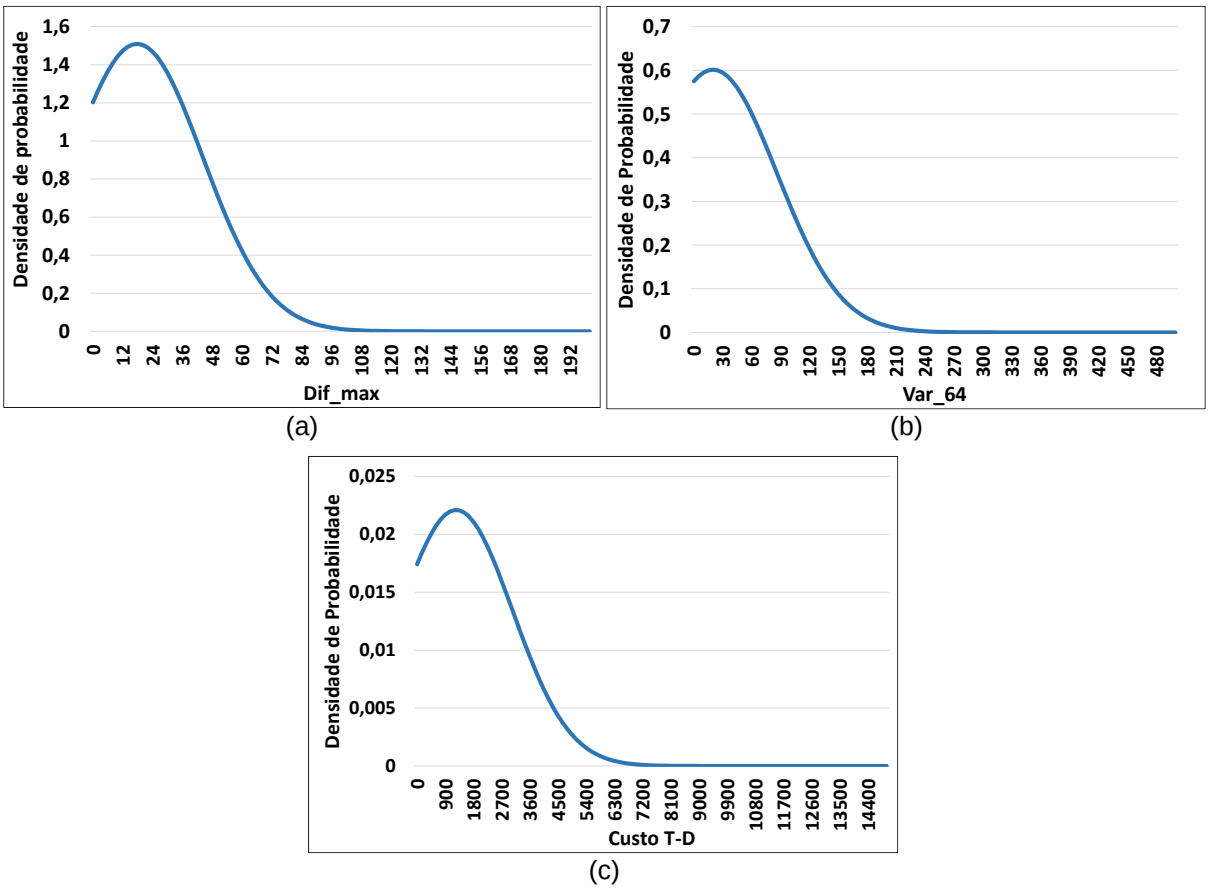


Figura 18 – Curva de densidade de probabilidade de UCs 64×64 que não foram divididas com a configuração AI.

Através da avaliação dos atributos foi possível concluir que somente os seguintes atributos foram relevantes para construir as árvores de decisão: *QP*, *Custo T-D*, *Var*, *Var_tam*, *Média* e *Dif_max*. Esses atributos foram selecionados no processo de definição de cada árvore de decisão pelo próprio algoritmo de construção das árvores que utiliza o ganho de informação (*Information Gain* – IG) durante o processo para definição dos atributos. O ganho de informação de cada atributo refere-se a diferença entre o cálculo de entropia para todo o conjunto de

dados e a entropia do subconjunto particionado para tal atributo que está sendo avaliado. Este processo é realizado para cada atributo do conjunto de dados.

A Tabela 2 apresenta a lista completa dos atributos utilizados nas árvores de decisão dos quadros I para cada tamanho de UC possível e os ganhos de informação (IG) obtidos por cada atributo utilizado. Os atributos *Var_32*, *Var_16*, *Var_8* e *Var_4* são as máximas variâncias de sub-blocos dentro da UC atual de tamanhos 32×32, 16×16, 8×8 e 4×4, respectivamente. Os demais atributos já foram detalhados anteriormente.

Tabela 2 – Atributos utilizados nas árvores de decisão para quadros I.

Atributo	**64×64**		**32×32**		**16×16**	
	Utilizado	**IG**	**Utilizado**	**IG**	**Utilizado**	**IG**
QP	×	0,089	×	0,123	×	0,039
Custo T-D	×	0,312	×	0,203	×	0,293
Var	×	0,395	×	0,17	×	0,027
Var_16	×	0,397		-		-
Var_8		-	×	0,171	×	0,025
Var_4		-		-	×	0,025
Dif_max		-		-	×	0,026
Média		-	×	0,11	×	0,061

A Figura 19 ilustra a árvore de decisão estática treinada para as UCs de tamanho 64×64 para os quadros I, onde as folhas identificadas pelos *labels* “S” e “N” correspondem às decisões de realizar ou não a divisão da UC, respectivamente. Os valores definidos para comparação de cada atributo são definidos com base nos dados de treinamento pelo algoritmo de aprendizado de máquina.

A árvore de decisão para as UCs de tamanho 64×64 possui uma profundidade de cinco níveis de decisão. As árvores de decisão para as UCs de tamanho 32×32 e 16×16 possuem uma estrutura similar a árvore de decisão criada para as UCs 64×64 e são compostas de profundidades de cinco e oito níveis de decisão, respectivamente. Os níveis de profundidade das árvores foram definidos visando alcançar o menor nível de profundidade com a maior acurácia possível (detalhados no Capítulo 6).

As árvores de decisão para as UCs 32×32 e 16×16 estão apresentadas no Apêndice A dessa dissertação.

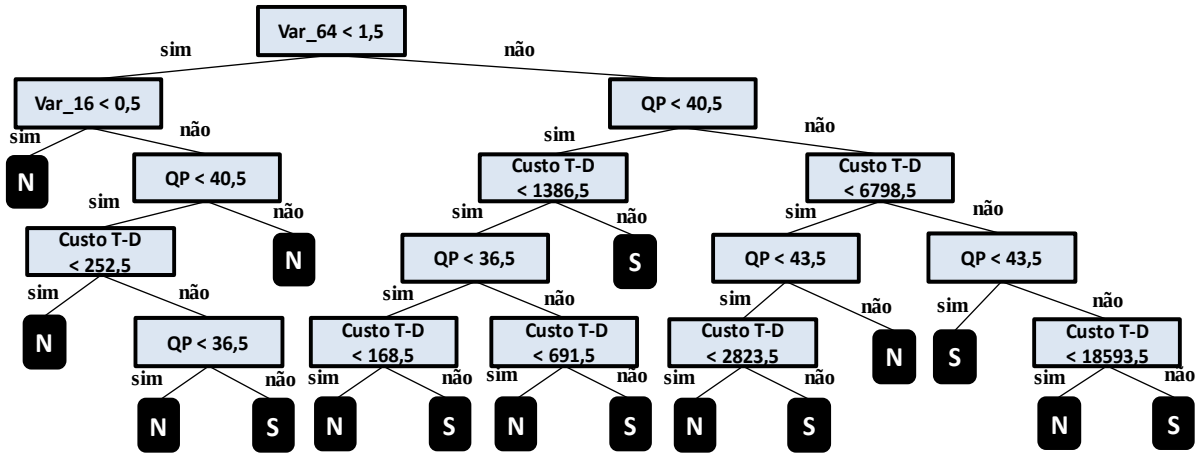


Figura 19 – Ilustração da árvore de decisão para UCs de tamanho 64×64 para quadros I.

5.2.2 Avaliação dos atributos e definição das árvores de decisão para quadros unidirecionais (P) e bidirecionais (B)

Seguindo o mesmo processo dos quadros I, foram analisados atributos que podem lidar de forma efetiva com a decisão de divisão das UCs para quadros P e B. Os mesmos atributos analisados nos quadros I também foram avaliados neste caso. No entanto, também foram avaliados novos atributos, já que os quadros P e B possuem várias características diferentes dos quadros I. Essas características diferem, principalmente, quando consideradas as dependências inter-quadros e inter-vistas. Por exemplo, os atributos que consideram o *RD-cost* durante a codificação fornecem informações importantes para definir se o processo de codificação da UC pode ser finalizado e, considerando o cenário de quadros P e B, essa informação pode ser extraída de diversas etapas da codificação, tal como ao utilizar o modo *Merge*/*SKIP* ou realizar a predição inter-quadros normalmente.

Portanto, além dos atributos utilizados no treinamento das árvores para quadros I, os seguintes atributos também foram analisados para treinamento das árvores de decisão para os quadros P e B:

- ***TD_MSM*** - referente ao *RD-cost* do modo *Merge/SKIP* (detalhes na Subseção 2.2.2). Este atributo indica a eficiência de codificação ao utilizar o modo *Merge/SKIP*, que geralmente alcança alta eficiência em regiões com baixo índice de movimentação ou em regiões homogêneas. Neste caso, geralmente são utilizados tamanhos de UCs maiores, o que ajuda na decisão de não realizar a divisão da UC.

- ***TD_DIS*** - relacionado ao *RD-cost* da ferramenta DIS (detalhes na Subseção 2.5.5). Apesar de quadros P e B permitirem a utilização das predições inter-quadros e inter-vistas, estes quadros também utilizam a predição intra-quadro para codificar algumas UCs. Baseado neste fato, verificar a eficiência de codificação ao utilizar o DIS na UC atual é uma boa alternativa, já que esta ferramenta foi desenvolvida para codificar de forma eficiente as regiões e existe baixa probabilidade de realizar a divisão da UC quando o valor do *TD_DIS* é baixo.
- ***TaxaIntra*** - relacionado ao *RD-cost* da UC atual codificada pela predição intra-quadro (Subseção 2.5.6) dividido pelo *RD-cost* da UC codificada pelo modo DIS. Esse atributo permite relacionar a eficiência de duas ferramentas que são utilizadas em diferentes cenários. A predição intra-quadro do 3D-HEVC tende a ser utilizada em UCs de regiões com mais detalhes e que tendem a utilizar tamanhos de UCs menores, quando comparada a ferramenta DIS.
- ***TaxaInter*** – é o RD-*cost* da UC atual codificada pela predição inter-quadros com PUs 2N×2N dividido pelo custo do modo *Merge/SKIP*. Este atributo é relevante, pois quando uma UC alcança melhor eficiência de codificação utilizando a predição inter-quadros, pode indicar que a UC pertence a regiões complexas ou com maior intensidade de movimento e, neste caso, as UCs tendem a ser divididas em tamanhos menores (CORRÊA, 2015).
- ***RelTaxa*** – é referente a diferença normalizada entre o *RD-cost* da predição inter-quadros com PUs 2N×2N e o *RD-cost* do modo *Merge/SKIP*. Este atributo parte da mesma ideia da *TaxaInter*. A Equação (1) descreve o cálculo deste atributo.

$$RelTaxa = \left|\frac{RD(2N\times 2N) - RD(MSM)}{RD(MSM)}\right| \quad (1)$$

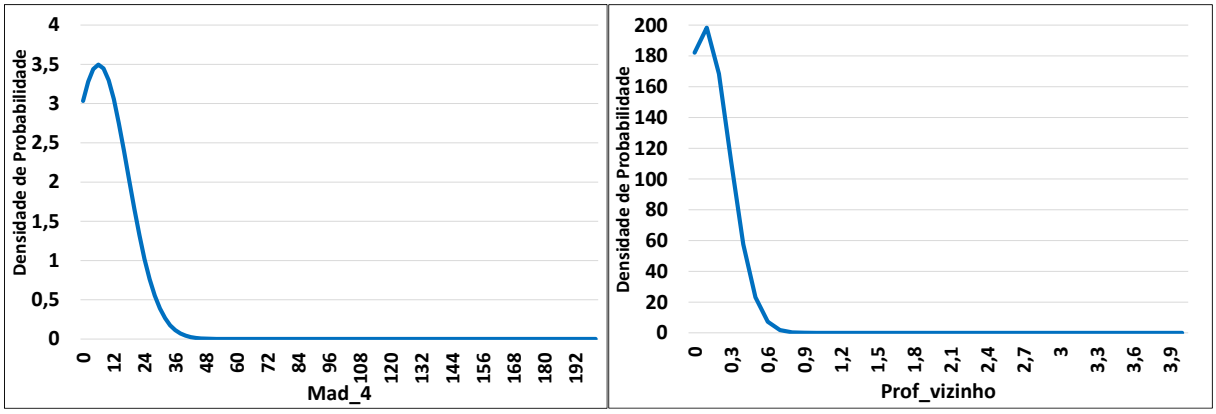
- ***Prof_vizinho*** - computa a média dos níveis de profundidade das *quadtrees* das CTUs vizinhas já codificadas. As seguintes CTUs são consideradas neste cálculo: (i) acima; (ii) esquerda; (iii) acima esquerda; (iv) acima direita; e (v) co-localizadas de ambas listas de referência

(temporalmente passada e futura). Este atributo explora as relações das *quadtrees* de CTUs vizinhas que tendem a ter uma alta correlação.

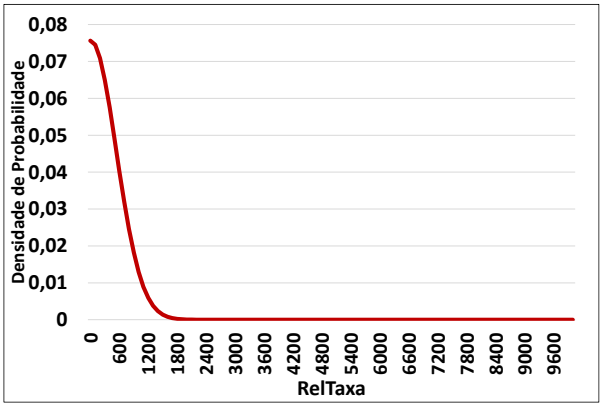
- ***SKIP*** - é um valor binário indicando se a UC atual foi codificada com o modo SKIP ou não. Em casos onde as UCs foram codificadas com o modo SKIP existe uma alta probabilidade da UC não ser dividida em tamanhos menores.
- ***DIS*** - é um valor binário que indica se a UC atual foi codificada com a ferramenta DIS ou não. Também neste caso, quando UCs são codificadas com o DIS, estas UCs tendem a não ser divididas em UCs de tamanhos menores.
- ***Mad*** - indica o máximo desvio médio absoluto de blocos menores dentro da UC atual sendo codificada. Esta informação estatística permite verificar a dispersão dos valores dentro da UC, e quando valores maiores deste atributo são encontrados, as UCs tendem a ser divididas.

A Figura 20 exemplifica as curvas de densidade de probabilidade considerando três atributos para UCs 64×64. A Figura 20 (a) apresenta a densidade de probabilidade das UCs não dividirem de acordo com o *Mad_4* (sub-blocos 4×4). A Figura 20 (b) ilustra a densidade de probabilidade das UCs não serem divididas de acordo com os valores do atributo *Prof_vizinho,* e a Figura 20 (c) apresenta a curva de densidade de probabilidade das UCs realizarem a divisão da UC de acordo com os valores do atributo *RelTaxa.* Por um lado, para os atributos *Mad_4* e *Prof_vizinho*, valores menores tendem a decidir por não realizar a divisão em tamanhos menores da UC que está sendo codificada. Por outro lado, valores baixos do atributo *RelTaxa* tendem a definir pela decisão de realizar a divisão da UC (optou-se por mostrar dessa forma para uma melhor visualização da curva).

A Tabela 3 apresenta a lista completa dos atributos utilizados nas árvores de decisão dos quadros P e B para as UCs de tamanho 64×64, 32×32 e 16×16, assim como os ganhos de informação (IG) de cada atributo. É importante notar que atributos, como *QP*, *Var_tam* e *Dif_max*, utilizados na definição das árvores de decisão dos quadros I, também foram utilizados na definição das árvores de decisão dos quadros P e B. Essas definições dos atributos foram realizadas pelo algoritmo de treinamento da árvore de decisão que utiliza o ganho de informação para a definição dos atributos.



(a) (b)



(c)

Figura 20 - Curva de densidade de probabilidade de UCs 64×64 com a configuração RA: (a) e (b) UCs não divididas; e (c) UCs divididas.

Tabela 3 – Atributos utilizados nas árvores de decisão para quadros P e B.

Atributo	64×64		32×32		16×16	
	Utilizado	IG	Utilizado	IG	Utilizado	IG
QP		-	×	0,032	×	0,022
Custo T-D	×	0,194	×	0,223	×	0,216
TD_MSM	×	0,198	×	0,182	×	0,154
TD_DIS		-	×	0,214		-
TaxaIntra	×	0,450	×	0,258	×	0,228
TaxaInter		-	×	0,194		-
RelTaxa	×	0,483	×	0,194	×	0,139
Prof_vizinho	×	0,493		-		-
SKIP	×	0,257	×	0,068	×	0,050
DIS		-	×	0,063		-
Var	×	0,269	×	0,209		-
Var_16	×	0,274		-		-
Mad_4	×	0,271		-		-
Dif_max	×	0,278		-		-

A Figura 21 apresenta a árvore de decisão treinada pelo algoritmo de

aprendizado de máquina para UCs 64×64 em quadros P e B. Esta árvore possui profundidade de cinco níveis de decisão. As árvores de decisão para UCs 32×32 e 16×16 possuem uma estrutura similar à apresentada para as UCs 64×64 e requerem uma profundidade de seis níveis de decisão. As árvores de decisão para UCs 32×32 e 16×16 estão apresentadas no Apêndice B dessa dissertação.

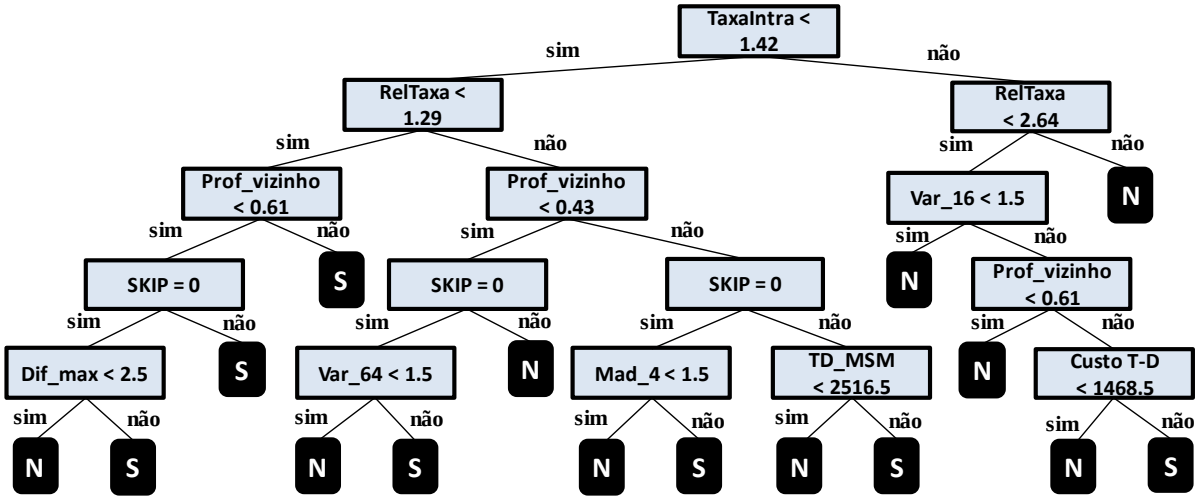


Figura 21 - Ilustração da árvore de decisão para UCs de tamanho 64×64 para quadros P e B.

É importante destacar que as seis árvores de decisão (três para quadros I e três para quadros P e B), propostas neste trabalho, causam um impacto adicional irrelevante (como cálculo da variância, por exemplo) considerando o esforço computacional do codificador 3D-HEVC, já que o treinamento é uma operação *offline* e a utilização das árvores de decisão geradas não aumenta o custo computacional do sistema.

6 RESULTADOS E DISCUSSÕES

Este capítulo está dividido em duas partes com os objetivos de (i) apresentar a metodologia utilizada para avaliar a solução proposta nesta dissertação, e (ii) discutir os resultados alcançados, com simulações realizadas em software, e comparar com trabalhos relacionados disponíveis na literatura.

6.1 Metodologia para as avaliações

Assim como em outras áreas de pesquisa, quando uma solução é proposta na área de codificação de vídeos, se faz necessária a realização de experimentos com alguns casos de testes para verificar a sua eficiência. No entanto, soluções focando na mesma área de pesquisa podem ser avaliadas com diferentes casos de testes, dificultando avaliar a contribuição e realizar comparações com trabalhos relacionados.

Para isso, no cenário de codificação de vídeos foram definidas as Condições Comuns de Teste (CCTs) (MULLER, 2014) que especificam os casos de testes e os cenários de avaliações. As CCTs para o 3D-HEVC englobam características significativas das configurações de codificação e utilizam diversas sequências de vídeos para testar diferentes cenários. Portanto, para avaliar novas soluções propostas para o codificador 3D-HEVC as CCTs devem ser utilizadas.

As CCTs detalham as especificações de avaliação considerando sequências de vídeos com e sem mapas de profundidade associados. Em ambos os casos, podem ser analisados os resultados para os cenários de codificação considerando duas ou três vistas.

Esta dissertação considera o cenário de codificação utilizando três vistas (com seus mapas de profundidade) para facilitar a comparação com trabalhos relacionados, já que grande parte dos trabalhos para redução do tempo de execução no 3D-HEVC usam este cenário. Além disso, os resultados para duas vistas podem ser derivados a partir dos resultados com três vistas.

As CCTs definem duas classes de resoluções para vídeos 3D e disponibilizam três sequências de teste com resolução de 1024×768 pixels e cinco com resolução 1920×1088 pixels. A Tabela 4 apresenta as sequências de teste utilizadas nas CCTs para avaliações, especificando o número de quadros para serem codificados, a taxa de amostragem dos quadros (*Frames per Second* – FPS)

e as vistas utilizadas considerando os cenários com duas e três vistas. A especificação dos números de vistas é necessária, pois as sequências geralmente são capturadas e disponibilizadas com mais do que três vistas. Mais detalhes das sequências de testes estão apresentados no Apêndice C.

Tabela 4 – Especificações das sequências de teste considerando duas ou três vistas.

Resolução	Sequência de teste	Quadros	FPS	2 vistas	3 vistas
1024×768	Balloons	300	30	1-3	1-3-5
	Kendo	300	30	1-3	1-3-5
	Newspaper1	300	30	2-4	2-4-6
1920×1088	GT_Fly	250	25	9-5	9-5-1
	Poznan_Hall2	200	25	7-6	7-6-5
	Poznan_Street	250	25	5-4	5-4-3
	Undo_Dancer	250	25	1-5	1-5-9
	Shark	300	30	1-5	1-5-9

Os valores de QP utilizados nas avaliações também são definidos pelas CCTs, onde diferentes pontos de operação são avaliados levando em consideração a taxa de compressão e a qualidade do vídeo. As sequências de teste devem ser avaliadas com os seguintes pares de valores de QPs para a vista base: (25, 34), (30, 39), (35, 42) e (40, 45). Cada par de valores de QPs são avaliados separadamente, ou seja, cada par de QPs indica uma execução diferente. A Tabela 5 apresenta diversas possibilidades dos valores de QPs dos mapas de profundidade em função do valor do QP utilizado para textura. QP_{v0} indica o valor do QP para vista base de textura e o QP_{d0} indica o valor do QP para vista base do mapa de profundidade. Em destaque estão os pares de QP definidos pela CCTs para a realização dos experimentos.

Tabela 5 – Valores dos QPs dos mapas de profundidade em função dos QPs de textura.

QP_{v0}	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
QP_{d0}	51	50	50	50	50	49	48	47	47	46	45	45	44	44	43	43	42	42	41	41	40	39	38	37	36	35	34

Os resultados obtidos são apresentados considerando a redução do tempo de execução na codificação dos mapas de profundidade e a eficiência de codificação das vistas sintetizadas. Para obter esses resultados, é necessário realizar a comparação com os resultados alcançados pelo 3D-HTM sem nenhuma modificação (com o fluxo RDO convencional) e, portanto, primeiramente são realizadas execuções com o 3D-HTM original (com QTL/QTPred desabilitado),

utilizando as CCTs. A seguir, as árvores de decisão foram inseridas no 3D-HTM e novas execuções considerando as CCTs foram realizadas.

Os resultados de redução do tempo de execução (RTE) são computados através da Equação (2),

$$RTE = \frac{To - Tm}{To} \times 100 \quad (2)$$

Onde To é o tempo de codificação do 3D-HTM original e Tm representa o tempo de codificação do 3D-HTM modificado.

Os resultados considerando a eficiência de codificação foram obtidos através da métrica *Bjontegaard Delta-rate* (*BD-rate*) (BJONTEGAARD, 2001). Esta métrica é frequentemente utilizada na área de codificação de vídeos e realiza uma relação entre a taxa de bits e a qualidade do vídeo codificado. Resultados de *BD-rate* positivos indicam que é necessária uma taxa de bits maior para representar o vídeo na mesma qualidade do vídeo de referência. Como esta dissertação aborda soluções para a codificação de mapas de profundidade, os resultados de *BD-rate* são apresentados apenas para as vistas sintetizadas.

É importante enfatizar que, assim como nas avaliações anteriores, as simulações neste capítulo são realizadas com ferramentas de redução do tempo de execução para *quadtree* dos mapas de profundidade (QTL/QTPred) desabilitadas.

As simulações foram realizadas utilizando duas configurações do codificador especificadas pelas CCTs. A configuração *All-Intra* (AI) (Subseção 6.2) foi utilizada para avaliar as árvores de decisão definidas para os quadros I, enquanto que a configuração *Random Access* (RA) (Subseção 6.3) foi utilizada para avaliar a solução com as árvores de decisão para todos os tipos de quadros (i.e., I, P e B).

6.2 Resultados em *All-Intra*

A Tabela 6 apresenta os resultados de acurácia obtidos para cada árvore de decisão treinada para os quadros I. As acurácias alcançadas foram 84,54%, 83,59% e 87,09% para os tamanhos de UCs 16×16, 32×32 e 64×64, respectivamente.

Tabela 6 – Acurácia das árvores de decisão para cada tamanho de UC em quadros I.

UC	Acurácia
16×16	84,54%
32×32	83,59%
64×64	87,09%

A Tabela 7 apresenta os resultados experimentais para as árvores de decisão definidas para os quadros I (“Árvores de decisão AI”), com o codificador sendo executado na configuração AI. A sequência de vídeo Kendo que foi utilizada para treinamento não foi utilizada para gerar o resultado final “média”, mas o resultado dessa sequência é apresentado no final da tabela para demonstrar que não ocorreu sobreajuste nos dados de treinamento.

A solução proposta foi capaz de alcançar uma redução no tempo de codificação variando de 37,3% até 63,4%, com uma média de 52,8%. Considerando apenas a codificação dos mapas de profundidade, o tempo de codificação foi reduzido em 59%. Esta redução expressiva tem impacto no *BD-rate* de apenas 0,18% (entre 0,04% e 0,62%).

Tabela 7 – Resultados experimentais com a configuração do codificador AI.

Vídeo	Árvores de decisão AI		(PENG, 2016)		(ZHANG, 2016)	
	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução
Balloons	0,14%	45,7%	1,2%	27,6%	0,29%	41%
Kendo	-	-	0,7%	26,3%	0,29%	39%
Newspaper_CC	0,11%	37,3%	1,1%	26,6%	-0,23%	36%
GT_Fly	0,06%	58,8%	0,3%	45,1%	0,20%	45%
Poznan_Hall2	0,62%	63,4%	1,3%	43,7%	0,33%	48%
Poznan_Street	0,12%	55,8%	1,4%	49,1%	1,16%	40%
Undo_Dancer	0,14%	56,5%	0,6%	49,1%	1,01%	39%
Shark	0,04%	51,9%	0,1%	33,7%	-	-
Média	0,18%	52,8%	0,8%	37,7%	0,44%	41%
Kendo	0,19%	49,8%	-	-	-	-

Esta variação de redução do tempo de codificação e de *BD-rate* ocorre, principalmente, por conta das diferentes características das sequências utilizadas nas avaliações. As sequências de resoluções maiores tendem a ser codificadas com

tamanhos de UCs maiores, por possuírem grandes regiões homogêneas e, portanto, realizando mais decisões antecipadas para finalizar o processo de codificação. A sequência Poznan_Hall2 é um exemplo de vídeo de alta resolução que foi capaz de alcançar a maior redução no tempo de execução ao custo de um maior impacto no *BD-rate* por ter uma estrutura de profundidade mais complexa. Em contrapartida, a sequência Newspaper_CC (de menor resolução), por exemplo, possui os mapas de profundidade com muitas distorções, o que dificulta a decisão de finalizar a avaliação dos modos de codificação de forma antecipada, alcançando uma menor redução do tempo de execução.

Além disso, as árvores de decisão AI foram comparadas com trabalhos relacionados que também visam reduzir o tempo de processamento das *quadtrees* dos mapas de profundidade em quadros I. Os resultados apresentados são para os trabalhos de (PENG, 2016) e (ZHANG, 2016) e podem ser visualizados na Tabela 7.

O método proposto por (PENG, 2016) é capaz de alcançar uma redução de tempo de execução média de 37,7% com um aumento médio no *BD-rate* das vistas sintetizadas de 0,8%. Em (ZHANG, 2016), uma redução do tempo de execução média de 41% foi obtida com um impacto no *BD-rate* de 0,44%. Portanto, a solução proposta nesta dissertação para redução do tempo na codificação de quadros I foi capaz de alcançar a maior redução do tempo de codificação quando comparado com os trabalhos relacionados. Além disso, esta solução alcança a menor degradação na eficiência de codificação entre os trabalhos relacionados, onde o *BD-rate* médio das vistas sintetizadas é 4,4 vezes melhor que (PENG, 2016) e 2,4 vezes melhor que (ZHANG, 2016).

A Figura 22 ilustra os resultados em porcentagem das UCs que não foram divididas (i.e., tamanhos menores não foram avaliados durante a codificação) de acordo com o tamanho da UC e o valor do QP de profundidade. Valores maiores de QPs levam aos maiores índices de decisão de não realizar a divisão da UC. Em média, 85% das UCs de tamanho 64×64 não foram divididas em tamanhos menores com QP=(40, 45). Este fato se justifica, pois este QP alcança elevadas taxas de compressão e tende a realizar a codificação das CTUs com tamanhos de UCs maiores. Já para o QP=(25, 34), a quantidade de UCs não divididas é menor, uma vez que valores menores de QP tendem a manter um maior nível de detalhes do vídeo durante a codificação e, para isso, é necessário utilizar UCs com tamanhos menores.

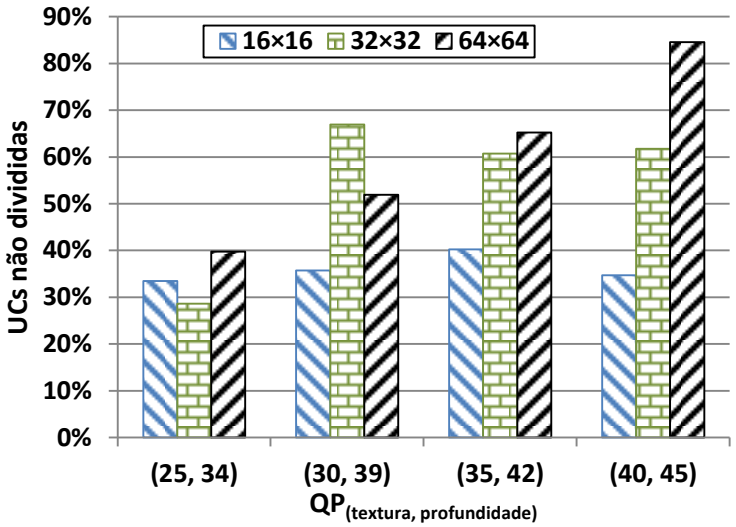
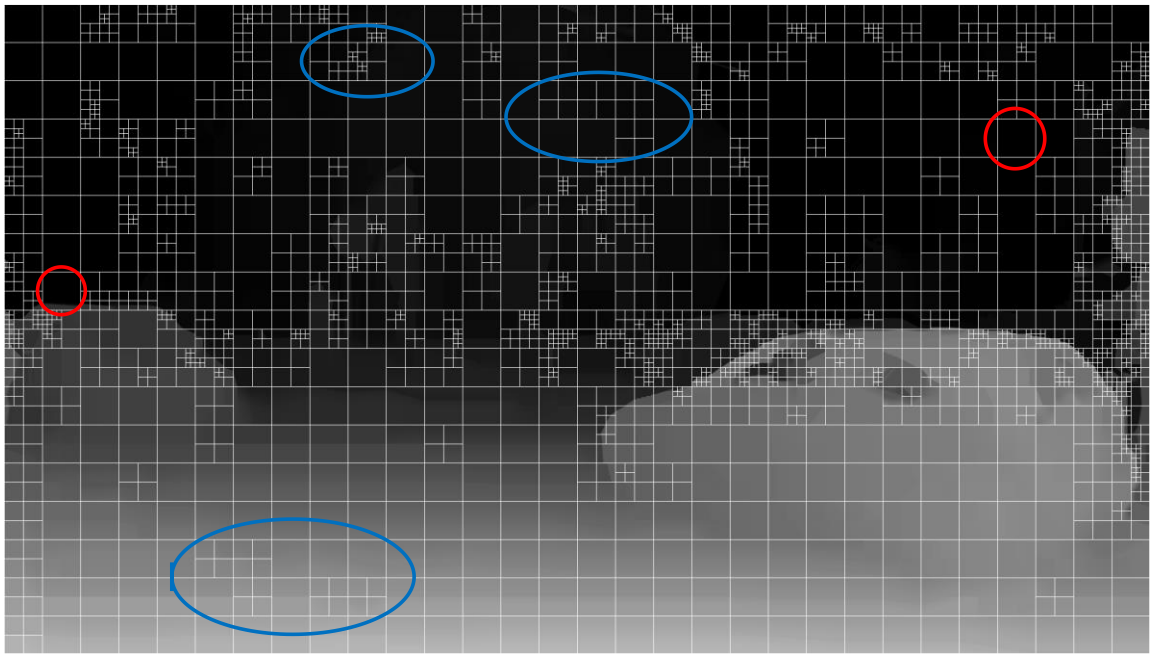
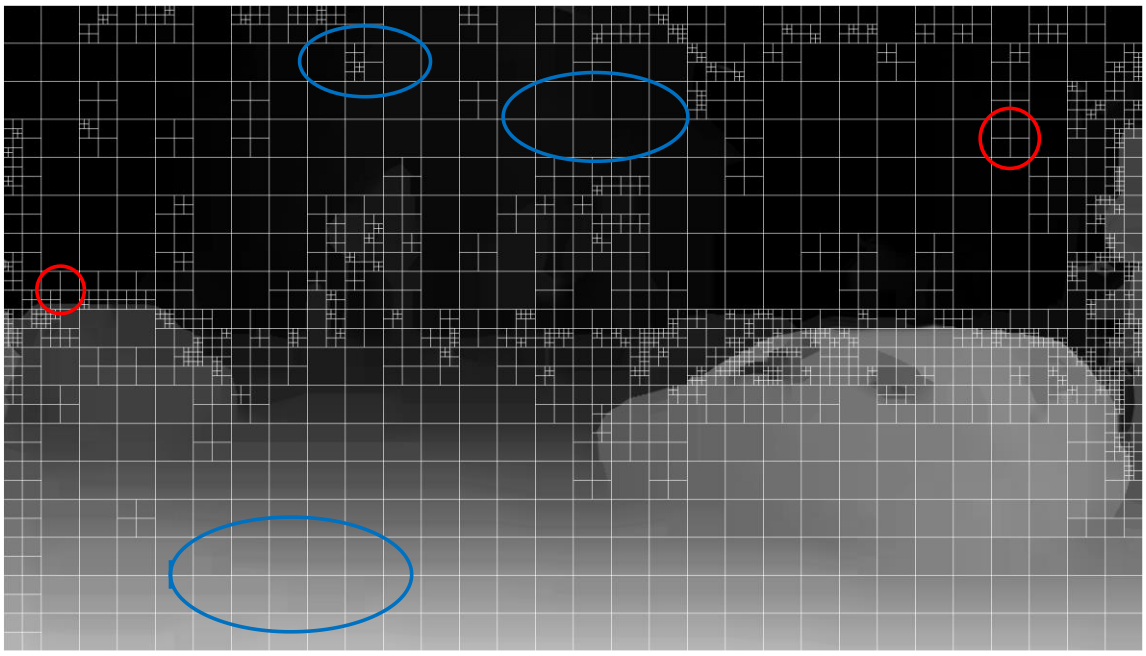


Figura 22 – Porcentagem de UCs que não foram divididas para a configuração AI de acordo com o tamanho e o valor do QP dos mapas profundidade.

Para uma melhor visualização do funcionamento da solução, a Figura 23 apresenta a distribuição de tamanhos de UCs do mapa de profundidade da sequência de vídeo *Poznan_Street* (segundo quadro da vista quatro), considerando a configuração AI com QP=(30, 39). A Figura 23 (a) ilustra a distribuição de tamanhos de UCs obtida com o 3D-HTM sem modificações e a Figura 23 (b) apresenta a distribuição de UCs alcançada pela solução proposta. Comparando as figuras é possível notar que mesmo utilizando a técnica com as árvores de decisão para reduzir o custo computacional, as decisões de divisões de UCs após a codificação possuem um comportamento semelhante, quando comparado ao 3D-HTM que realiza uma avaliação mais custosa computacionalmente (avalia todas possibilidades de UCs). Em alguns casos a solução proposta decide por utilizar mais divisões do que a decisão do 3D-HTM sem modificações. Contudo, em outros casos, a solução proposta decide por utilizar um número menor de divisões (empregando tamanhos maiores de UCs que o 3D-HTM) e, consequentemente, diminuindo a quantidade de informações necessária para gerar o *bitstream*. Os círculos em azul na Figura 23 (a) e (b) apresentam exemplos onde ocorreram menos divisões aplicando a solução proposta, e os círculos em vermelho apresentam a situação oposta.



(a) Codificado com o 3D-HTM sem modificações



(b) Codificado com a solução proposta

Figura 23 – Distribuição de tamanhos de UCs do mapa de profundidade da sequência de vídeo *Poznan_Street* codificado pelo (a) 3D-HTM sem modificações e a (b) solução proposta.

6.3 Resultados em *Random Access*

A Tabela 8 apresenta os resultados de acurácia para as árvores de decisão treinadas para os quadros P e B, onde as árvores para as UCs 16×16, 32×32 e 64×64 obtiveram acurácias de 82,13%, 83,08% e 91,65%, respectivamente.

Tabela 8 – Acurácia das árvores de decisão para cada tamanho de UC em quadros P e B.

UC	Acurácia
16×16	82,13%
32×32	83,08%
64×64	91,65%

A Tabela 9 apresenta os resultados da solução proposta, composto por três árvores de decisão responsáveis pela codificação dos quadros I e três para os quadros P e B. Estes resultados foram obtidos considerando a redução do tempo de execução do codificador configurado em RA e o impacto no *BD-rate* das vistas sintetizadas. Neste caso, os resultados da sequência de treinamento (Kendo) também foram desconsiderados para obter o resultado final, mas foram apresentados na última linha da tabela para demonstrar que não houve sobreajuste nos dados de treinamento.

Tabela 9 – Resultados experimentais com a configuração do codificador em RA.

Vídeo	Solução proposta		(MORA, 2014)		(LEI, 2016)	
	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução	Vistas sintetizadas (*BD-rate*)	Redução do Tempo de Execução
Balloons	0,06%	47,7%	2,03%	48,2%	4,32%	39,7%
Kendo	-	-	1,65%	48,5%	1,20%	33,9%
Newspaper_CC	0,05%	48,1%	1,40%	52,3%	3,43%	43,0%
GT_Fly	0,07%	52,2%	0,33%	50,3%	0,87%	45,1%
Poznan_Hall2	0,39%	59,4%	2,01%	58,6%	3,31%	43,3%
Poznan_Street	0,10%	57,0%	0,05%	56,5%	1,22%	45,4%
Undo_Dancer	0,55%	53,3%	0,37%	50,7%	1,67%	48,3%
Shark	0,01%	51,4%	0,35%	50,8%	0,48%	33,3%
Média	0,18%	52,7%	1,02%	52,0%	2,06%	41,5%
Kendo	0,03%	47,1%	-	-	-	-

A solução proposta foi capaz de alcançar uma redução média do tempo de codificação de 52,7%, com um impacto médio de apenas 0,18% no *BD-rate* das vistas sintetizadas. A solução proposta apresenta uma variação entre 0,01% e 0,55% no *BD-rate* e reduz entre 47,7% e 59,4% o tempo de execução, considerando o cenário de avaliação das CCTs. Além disso, considerando apenas a codificação dos mapas de profundidade, a solução é capaz de alcançar uma redução média de

tempo de codificação de 68%.

Neste caso também ocorre uma pequena variação entre os resultados, que pode ser notada pelas diferentes classes de resolução, onde as cinco sequências com resoluções maiores (GT_Fly, Poznan_Hall2, Poznan_Street, Undo_Dancer e Shark) alcançaram as maiores reduções de tempo de execução devido a elevada utilização de UCs de tamanhos maiores para a codificação, já que tendem a possuir grandes regiões homogêneas.

A Tabela 9 também compara a solução proposta com os trabalhos de (MORA, 2014) e (LEI, 2016). O método estado-da-arte proposto por (MORA, 2014) (incorporado no 3D-HTM) e o trabalho em (LEI, 2016) são capazes de alcançar, em média, uma redução de tempo de codificação de 52,0% e 41,5%, com um aumento no *BD-rate* de 1,02% e 2,06%, respectivamente. Portanto, estes resultados demonstram que a solução proposta nesta dissertação é capaz de fornecer um compromisso mais eficiente para codificação dos mapas de profundidade no 3D-HEVC, com um impacto aproximadamente 5,6 vezes menor no *BD-rate* das vistas sintetizadas. Além disso, a Tabela 9 demonstra que a solução utilizando as árvores de decisão é capaz de trabalhar em cenários com diferentes características e apresentar resultados de *BD-rate* e redução de tempo de execução com menor variação, quando comparado aos trabalhos relacionados.

O baixo impacto no *BD-rate*, em comparação aos trabalhos relacionados, deve-se principalmente a uma maior abrangência de parâmetros de codificação e características extraídas das UCs codificadas que são utilizadas nas definições das árvores de decisão. Além disso, a utilização da solução especializada para quadros I permite uma elevada eficiência na codificação com baixa propagação de erros para os demais quadros.

A Figura 24 apresenta a porcentagem de UCs que não realizaram a divisão, considerando a configuração RA de acordo com o tamanho da UC e o valor do QP dos mapas de profundidade. Novamente, uma grande quantidade de UCs não foi dividida quando os mapas de profundidade são codificados com valores elevados de QPs (QP=(40, 45)). Para UCs de tamanho 64×64 codificadas com o QP=(40, 45) é observado que aproximadamente 97% das UCs não realizam a divisão para tamanhos menores. Analisando o QP=(25, 34), é possível notar que mais de 60% das UCs não foram divididas, considerando os tamanhos de UC 16×16, 32×32 e 64×64.

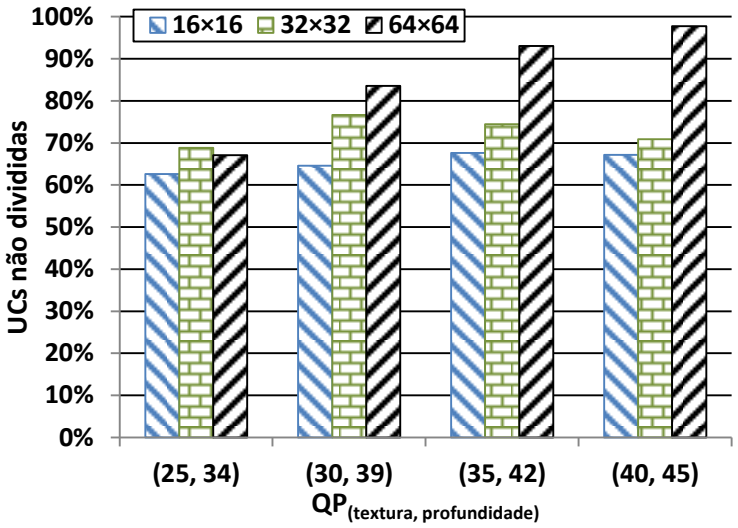
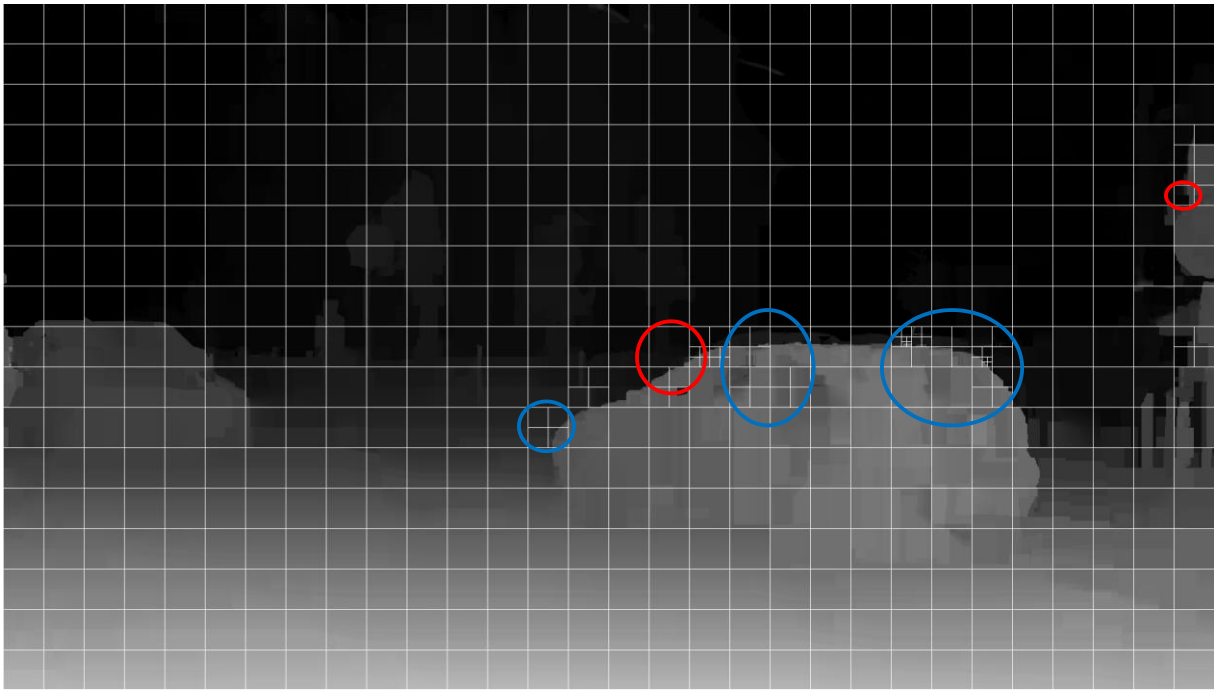
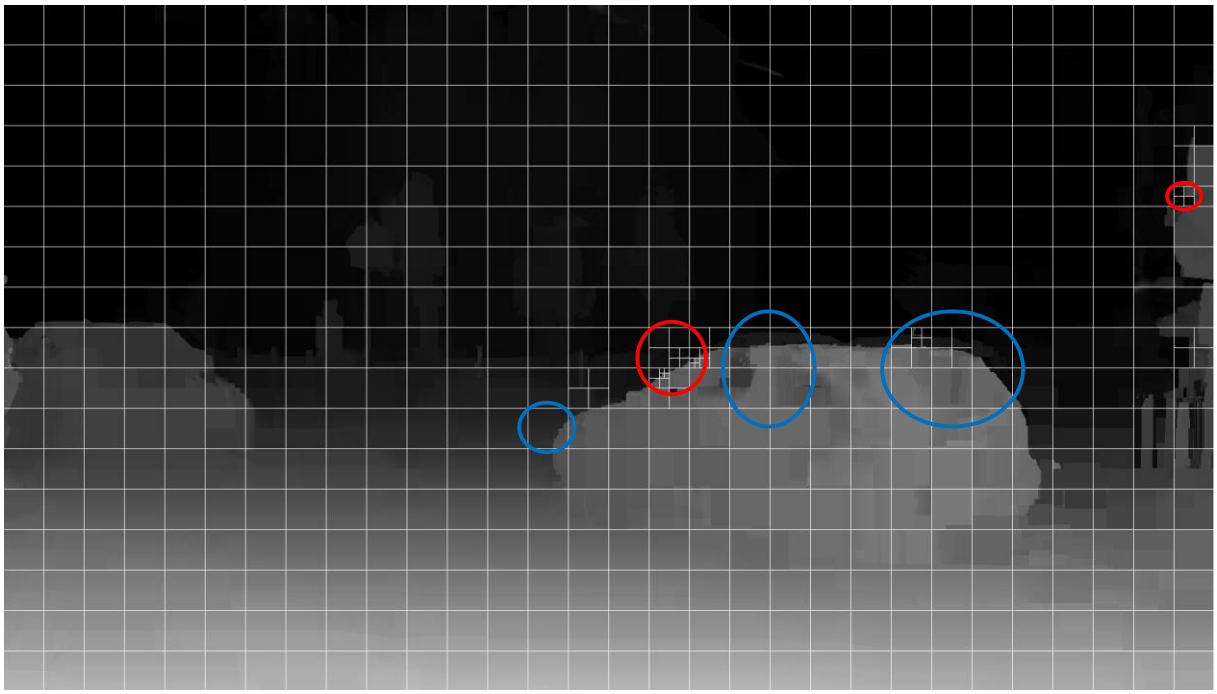


Figura 24 - Porcentagem de UCs que não foram divididas para a configuração RA de acordo com o tamanho e o valor do QP dos mapas de profundidade.

A Figura 25 (a) e a Figura 25 (b) ilustram a distribuição de tamanhos de UCs para o mapa de profundidade da sequência de vídeo *Poznan_Street* (sétimo quadro da vista quatro) na configuração RA com QP=(30, 39), considerando a codificação com o 3D-HTM sem modificações (com QTL/QTPred habilitado) e com a inserção da solução proposta no 3D-HTM.



(a) Codificado com o 3D-HTM sem modificações



(b) Codificado com a solução proposta

Figura 25 – Distribuição de tamanhos de UC do mapa de profundidade da sequência de vídeo *Poznan_Street* codificado pelo (a) 3D-HTM sem modificações e a (b) solução proposta.

É possível notar que, novamente, utilizando a solução desenvolvida para reduzir o custo computacional do codificador as divisões das UCs codificadas são similares, quando comparado as divisões obtidas pelo 3D-HTM sem modificações. No entanto, existem casos em que a solução proposta utiliza tamanhos menores de UCs (destacado em círculos vermelhos) e, a maior parte, de casos onde a solução proposta realiza menos divisões das UCs, resultando em um número menor de informações para geração do *bitstream*. Além disso, é possível notar que as predições inter-quadros e inter-vistas são capazes de alcançar resultados eficientes sem realizar muitas divisões nas UCs, ao contrário do que ocorreu na subseção anterior com os quadros I.

Ambos os experimentos (com configuração AI e RA) demonstraram que as árvores de decisão propostas são capazes de alcançar uma elevada redução de tempo de codificação, removendo a avaliação de tamanhos menores de UCs e mantendo a eficiência de codificação considerando a taxa de bits e a qualidade do vídeo. Além disso, para ambas configurações do codificador, a solução proposta é capaz de superar os resultados de redução de tempo de execução e *BD-rate* dos trabalhos relacionados encontrados na literatura.

7 CONCLUSÕES E TRABALHOS FUTUROS

O 3D-HEVC herda as estruturas e as ferramentas de codificação do HEVC para a codificação de vídeos 3D e é capaz de alcançar resultados extremamente eficientes. Esta eficiência também está diretamente relacionada ao fato de incorporar o formato de dados MVD, que inclui mapas de profundidade durante o processo de codificação. Estes mapas de profundidade são capazes de fornecer informações geométricas da cena e estão associados aos quadros de textura. A utilização do formato MVD reduz o problema da quantidade de dados necessários para transmissão em um sistema 3D, pois permite reduzir o número de vistas capturadas/codificadas e realizar a síntese de vistas intermediárias no decodificador através da interpolação de vistas de textura com mapas de profundidade.

Os mapas de profundidade não são exibidos para os usuários, mas são cruciais para o processo de síntese de vistas. Todavia, erros durante a codificação desses dados podem ocasionar em distorções na estrutura dos vídeos 3D. Baseado neste fato, além das ferramentas desenvolvidas para os dados de textura o 3D-HEVC também conta com ferramentas específicas para a codificação dos mapas de profundidade. Essas ferramentas são eficientes para preservar as informações dos mapas de profundidade durante a codificação e fornecer vistas sintetizadas de alta qualidade no entanto, estas ferramentas acabam elevando o custo computacional do codificador.

Nesta dissertação foi desenvolvido uma solução para redução do tempo da codificação dos mapas de profundidade baseado em *data mining* e árvores de decisão, focando nas *quadtrees* dos mapas de profundidade do 3D-HEVC para evitar a avaliação tradicional do 3D-HTM utilizando o RDO. A solução é dividida em duas partes: (i) árvores de decisão especializadas para quadros I; e (ii) árvores de decisão especializadas para quadros P e B.

Diversos atributos de codificação e variáveis internas foram coletados para o processo de *data mining*, tentando extrair características que possam indicar a decisão de divisão das UCs. Após, foi realizada uma avaliação para verificar quais foram os atributos mais relevantes para cada tamanho de UC. Para ajudar no pré-processamento dos dados e no processo de treinamento *offline* das árvores de decisão, foi utilizado o software WEKA.

A solução possui três árvores de decisão estáticas (para as UCs de tamanho

16×16, 32×32 e 64×64) responsáveis pela decisão de divisão das UCs de quadros I e três árvores de decisão estáticas especializadas na decisão de divisão das UCs de quadros P e B. Foram realizadas diversas análises, definição de atributos e treinamento das árvores de decisão considerando os quadros I e os quadros P e B.

A solução proposta nesta dissertação foi avaliada no 3D-HTM 16.0 utilizando as Condições Comuns de Testes para o 3D-HEVC. Primeiramente foi avaliada de forma isolada a solução para os quadros I, utilizando a configuração do codificador *All-Intra*. Os resultados para essa solução apresentaram uma redução média de tempo de codificação de 52,4%, considerando o tempo de execução dos dados de textura e mapas de profundidade, ao custo de um acréscimo no *BD-rate* das vistas sintetizadas de apenas 0,18%. Somado a isso, foram realizadas comparações com trabalhos relacionados focando na mesma etapa da codificação e a solução proposta com a utilização de árvores de decisão superou os trabalhos relacionados tanto em redução de tempo de execução quanto em *BD-rate*.

Por fim, as seis árvores de decisão foram implementadas no 3D-HTM e avaliadas na configuração do codificador *Random Access*. Os resultados demonstraram que a solução proposta reduz em média 52,7% o tempo da codificação, considerando o tempo de execução dos dados de textura e dos mapas de profundidade. Quando consideramos apenas o tempo de execução da codificação dos mapas de profundidade, ele apresenta uma redução de tempo de execução média de 68%. Esta expressiva redução no tempo de codificação gera um impacto médio de apenas 0,18% no *BD-rate* das vistas sintetizadas. Neste caso também foram realizadas comparações com trabalhos relacionados e a solução com árvores de decisão superou os trabalhos encontrados na literatura.

Através dessas avaliações, foi possível concluir que a solução proposta nesta dissertação alcança resultados significativos de redução de tempo na codificação dos mapas de profundidade com um impacto mínimo no *BD-rate* das vistas sintetizadas. Além disso, a solução não insere esforço computacional significativo no codificador, já que o treinamento é realizado apenas uma única vez, e de forma *offline*.

Além do que foi apresentado nesta dissertação, o 3D-HEVC ainda possui um vasto espaço de exploração na codificação dos mapas de profundidade. Novos modos para predição podem ser modificados ou desenvolvidos visando reduzir a alta complexidade causada pelos modos DMMs, por exemplo, ou melhorar a codificação

diminuindo as perdas de informação para que também melhore a qualidade das vistas sintetizadas. Além disso, o desenvolvimento de soluções de arquiteturas de hardware para vídeos 3D é um tópico interessante. Tipicamente, para atingir o desempenho em tempo real, são necessárias arquiteturas de hardwares dedicadas e algoritmos eficientes para redução do custo computacional no processo de codificação com isso, as arquiteturas podem contemplar os algoritmos desenvolvidos nesta dissertação.

REFERÊNCIAS

APTÉ, C., WEISS, S. “Data mining with decision trees and decision rules”, **Future generation computer systems (FGCS)**, v. 13, n. 2, pp. 197-210, Nov. 1997.

BJONTEGAARD, G. “**Calculation of average PSNR differences between RD Curves**”, VCEG-M33, ITU-T SG16/Q6 VCEG, 13th VCEG Meeting: Austin, USA, Abril de 2001.

BRUNK, C., PAZZANI, M. “An investigation of noise-tolerant relational concept learning algorithms”, in Proceedings of the **International Workshop on Machine Learning (MLSP)**, pp. 389-393, 1991.

CHEN, Y.; WANG, Y. UGUR, K., HANNUKSELA, M. LAINEMA, J.; GABBOUJ, M. “The Emerging MVC Standard for 3D Video Services”, in Proceedings of the **EURASIP Journal on Advances in Signal Processing (EURASIP)**. pp. 1-13, Mar. 2008.

CHEN, Y., TECH, G., WEGNER, K., YEA, S., “**Test Model 11 of 3D-HEVC and MV-HEVC**”, document JCT3V-K1003 of JCT-3V, Geneva, CH, 2015.

CHENG, Y., et al. “An H.264 Spatio-Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video”, in Proceedings of the **IEEE International Symposium on Circuits and Systems (ISCAS)**, pp. 880-883, 2009.

CORREA, G., ASSUNCAO, P., AGOSTINI L., CRUZ, L. “Fast HEVC Encoding Decisions Using Data Mining”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 25, n. 4, pp. 660-673, Oct. 2015.

FU, C.; ALSHINA, E.; ALSHIN, A.; HAUNG, Y.; CHEN, C.; TSAI, C.; HSU, C.; LEI, S.; PARK, J.; HAN, W. “Sample Adaptive Offset in HEVC Standard”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 2, pp. 1755-1764, Oct. 2012.

FU, C., ZHANG, H., SU, W., TSANG, S., CHAN, Y. “Fast Wedgelet Pattern Decision for DMM in 3D-HEVC”, in Proceedings of the **IEEE International Conference on Digital Signal Processing (DSP)**, pp. 477-481, 2015.

GHANBARI, M. “**Standard Codecs**: **Image Compression to Advanced Video Coding**”, [S.l.]: Institution Electrical Engineers, 2003.

GUO, X.; GAO, W.; ZHAO, D. “Motion vector prediction in multiview video coding”, Proceedings of the **IEEE International Conference on Image Processing (ICIP)**, pp. II-337, Sept. 2005.

HALL, M., et al. “**The WEKA data mining software: an update**”, SIGKDD Explorations, v. 11, n. 1, pp. 10-18, Jun. 2009.

HARRIS C., and SATEPHENS M. J. “A combined corner and edge detector”, in Proceedings of the **Alvey Vision Conference**, pp. 147-152, 1988.

JABALLAH, S., LARABI, M., TAHAR, J. “Heuristic Inspired Search Method for Fast Wedgelet Pattern Decision in 3D-HEVC”, in Proceedings of the **European Workshop on Visual Information Processing (EUVIP)**, pp. 1-6, 2016.

JAGER, F. “Depth-based Block Partitioning for 3D Video Coding”, in Proceedings of the **International Picture Coding Symposium (PCS)**, pp. 410-413, 2013.

JCT-VC Editors, “**Recommendation ITU-T H.265**”, Series H: Audiovisual and Multimedia Systems Infrastructure of Audiovisual Services – Coding of moving video, 2013.

JCT-3V, “**JCT-3V DOCUMENT MANAGEMENT SYSTEM**”, Disponível em: <http://phenix.int-evry.fr/jct2/>, acesso em Nov. de 2016.

KAUFF, P., et al. “Depth Map Creation and Image-based Rendering for Advanced 3DTV Services Providing Interoperability and Scalability”, **Signal Processing: Image Communication (SPIC)**, v. 22, n. 2, pp. 217-234, Feb. 2007.

KIM, M., LING, N., SONG, L. “Fast Single Depth Intra Mode Decision for Depth Map Coding in 3D-HEVC”, in Proceedings of the **IEEE International Conference on Multimedia & Expo Workshops (ICMEW)**, pp. 1-6, 2015.

LAINEMA, J. et al. “Intra Coding of the HEVC Standard”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, pp. 1792-1801, Dec. 2012.

LEE, J., PARK, M., KIM, C. “**3D-CE1: Depth Intra Skip (DIS) Mode**”, document JCT3V-K0033, Geneva, Switzerland, Feb. 2015.

LEI, J., DUAN, J., WU, F. “Fast Mode Decision Based on Grayscale Similarity and Inter-View Correlation for Depth Map Coding in 3D-HEVC”, **IEEE Transactions on Circuits and Systems (TCSVT)**, (online first), v. PP, n. 99, Oct. 2016.

LI, X.; ZHANG, L.; CHEN, Y. “ADVANCED RESIDUAL PREDICTION IN 3D-HEVC”, **in Proceedings of the IEEE International Conference on Image Processing (ICIP)**, pp. 1747-1751, 2013.

LIU, H; CHEN, Y. “Generic segment-wise DC for 3D-HEVC depth intra coding”, **in Proceedings of the IEEE International Conference on Image Processing (ICIP)**, pp. 3219-3222, 2014.

MA, S.; WANG, Y.; ZHU, C.; LIN, Y.; ZHENG, J. “Reducing Wedgelet Lookup Table Size with Down-Sampling for Depth Map Coding in 3D-HEVC”, in Proceedings of the **IEEE International Workshop on Multimedia Signal Processing (MMSP)**, pp. 1-5, 2015.

MA, S.; WANG, S.; GAO, S. “Low Complexity Adaptive View Synthesis Optimization in HEVC Based 3D Video Coding”, **IEEE Transactions on Multimedia (TMM)**, v. 16, n. 1, Jan. 2014.

MERKLE, P.; SMOLIC, A.; MULLER, K.; WIEGAND, T. “Multi-view Video Plus Depth Representation and Coding”, in Proceedings of the **IEEE International Conference on Image Processing (ICIP)**, pp. I - 201-I - 204, 2007.

MERKLE, P., MULLER, K., MARPE, D., WIEGAND, T. “Depth Intra Coding for 3D Video based on Geometric Primitives”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 26, n. 3, pp. 570-582, Feb. 2015.

MOBILE-3DTV. “**Mobile-3DTV Solid Video Database**”, Disponível em: <http://sp.cs.tut.fi/mobile3dtv/stereo-video/>. Acesso em 23 de janeiro de 2018.

MORA, E.; JUNG, J.; CAGNAZZO, M.; PESQUET-POPESCU, B. “Initialization, limitation and predictive coding of the depth and texture quadtree in 3D-HEVC”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 24, n. 9, pp. 1554-1565, Sep. 2014.

MULLER, K. et al. “3D High-Efficiency Video Coding for Multi-View Video and Depth Data”, **IEEE Transactions on Image Processing (TIP)**, v. 22, n. 9, pp. 3366-3378, Set. 2013.

MULLER, K., VETRO, A. “**Common Test Conditions of 3DV Core Experiments**”, ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, Jan. 2014.

NORKIN, A.; BJONTEGAARD, G.; FULDESETH, A.; NARROSCHKE, M. IKEDA, M.; ANDERSSON, K.; ZHOU, M.; VAN DER AUWERA, V. “HEVC Deblocking Filter”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, pp. 1746-1754, Dec. 2012.

PENG, K., CHIANG, J., LIE, W. “Low complexity depth intra coding combining fast intra mode and fast CU size decision in 3D-HEVC”, in Proceedings of the **IEEE International Conference on Image Processing (ICIP)**, pp. 1126-1130, 2016.

QUINLAN, J. “**Simplifying decision trees**”, Int. J. Man-Mach. Stud., v. 27, n. 3, pp. 221-234, Set. 1987.

QUINLAN, J. “**C4.5: Programs for Machine Learning**”, Morgan Kaufmann Publishers, 1993.

RICHARDSON, I. “**The H.264 Advanced Video Compression Standard**”, 2nd ed. Chichester: John Wiley and Sons, 2010.

ROSEWARNE, C.; BROSS, B.; NACCARI, M.; SHARMAN, K.; SULLIVAN, G. **High Efficiency Video Coding (HEVC) Test Model 16 (HM 16)**: Improved Encoder Description. [S.l.]: Joint Collaborative Team on Video Coding, 2015.

SALDANHA, M., SANCHEZ, G., ZATT, B., PORTO, M., AGOSTINI, L. “Complexity Reduction for 3D-HEVC Depth Maps Coding”, in Proceedings of the **IEEE International Symposium on Circuits and Systems (ISCAS)**, pp. 621-624, 2015.

SANCHEZ, G. “**DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC**”, 93 f. Dissertação (Mestrado em Computação) – Programa de Pós Graduação em

Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2014.

SANCHEZ, G., SALDANHA, M., BALOTA, G., ZATT B., PORTO, M., AGOSTINI, L. “Complexity Reduction for 3D-HEVC Depth Maps Intra-Frame Prediction Using Simplified Edge Detector Algorithm”, in Proceedings of the **IEEE International Conference on Image Processing (ICIP)**, pp. 3209-3213, 2014b.

SULLIVAN, G.; WIEGAND, T. “Rate-distortion optimization for video compression”, **IEEE Signal Processing Magazine**, v. 15, n. 6, pp. 74 – 90, Nov. 1998.

SULLIVAN, G.; OHM, J. HAN, W.; WIEGAND, T. “Overview of the High Efficiency Video Coding (HEVC) Standard”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 22, n. 12, pp. 1649-1668, Sep. 2012.

SULLIVAN, G., BOYCE, J., CHEN, Y., OHM, J., SEGALL, C., VETRO, A. “Standardized Extensions of High Efficiency Video Coding (HEVC)”, **IEEE Journal of Selected Topics in Signal Processing (J-STSP)**, v. 7, n. 6, pp. 1001-1016, Dec. 2013.

TECH, G. et al. “3D video coding using the synthesized view distortion change”, Picture Coding Symposium (PCS), pp. 25-28, 2012.

TECH, G.; et al. “Overview of the Multiview and 3D Extensions of High Efficiency Video Coding”, **IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)**, v. 26, n.1, pp. 35-49, Sep. 2015.

VETRO, A; SULLIVAN, G. “Overview of the Stereo and Multiview Video Coding Extension of the H.264/MPEG-4 AVC Standard”, **Proceedings of the IEEE**, v. 99, n. 4, Apr. 2011.

YASAKETHU, S.; FERNANDO, W.; KAMOLRAT, B.; KONDOZ, A. “Analyzing Perceptual Attributes of 3D Video”, **IEEE Transactions on Consumer Electronics (TCE)**, v. 55, n. 2, May 2009.

ZHANG, H., CHAN, Y., FU, C., TSANG, S., SIU, W. “Quadtree Decision for Depth Intra Coding in 3D-HEVC by Good Feature”, in Proceedings of the **International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 1481-1485, 2016.

ZHANG, H., FU, C., CHAN, Y., TSANG, S., SIU, W. “Probability-based Depth Intra Mode Skipping Strategy and Novel VSO Metric for DMM Decision in 3D-HEVC”, **IEEE Transactions on Circuits and Systems (TCSVT)**, (online first), v. PP, n. 99, 2016.

ZHANG, X. et al., “3D-CE2 related: A texture-partition-dependent depth partition for 3D-HEVC”, JCT3V-G0055, 2014.

ZHAO, X. et al., “**CE6.h related: Simplified DC predictor for depth intra**”, Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) document JCT3V-D0183, 4th JCT-3V Meeting: Incheon, Korea, Apr. 2013.

Apêndice A – Árvores de decisão para UCs 32×32 e 16×16 em quadros I

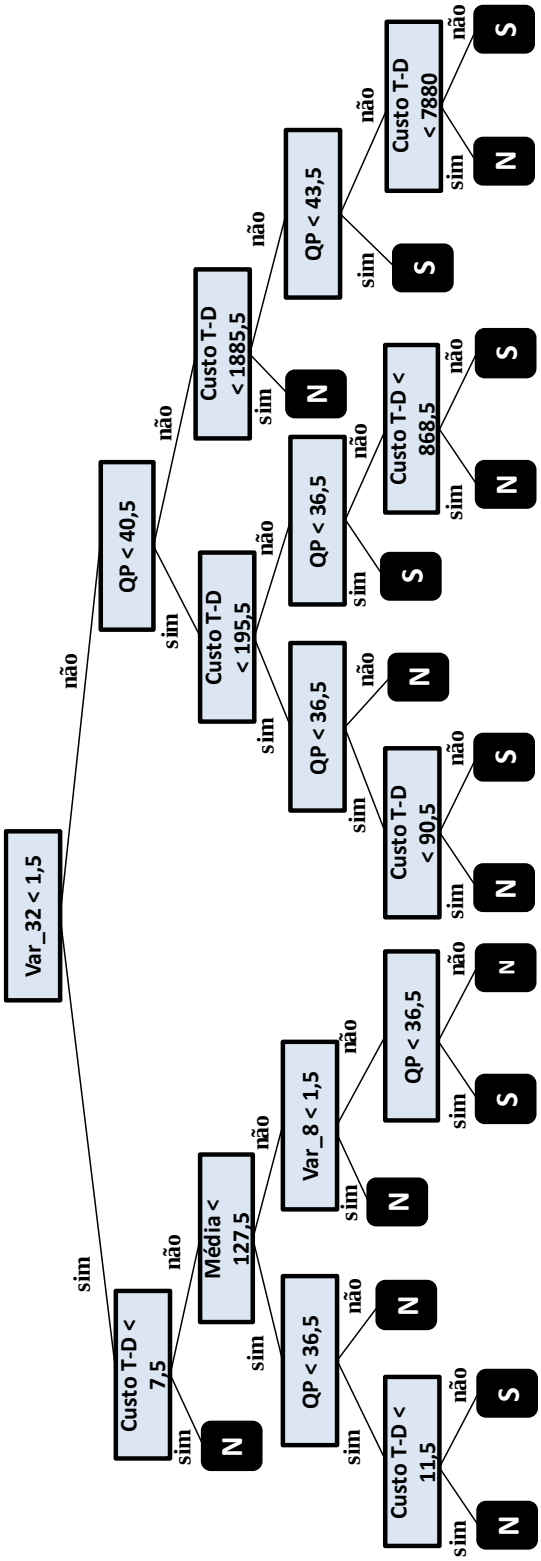


Figura 26 – Ilustração da árvore de decisão para UCs de tamanho 32×32 para quadros I.

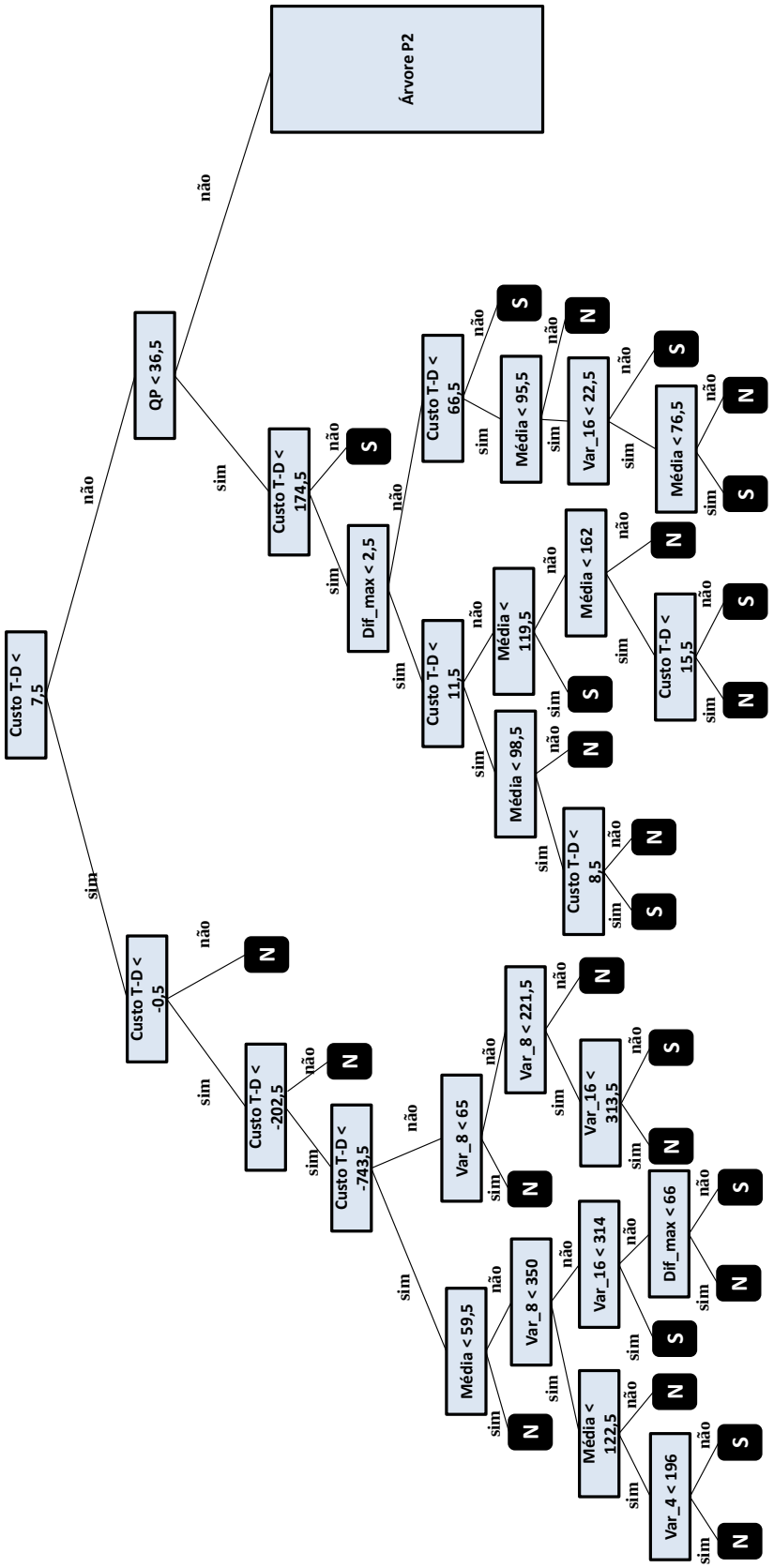


Figura 27 – Ilustração da árvore de decisão para UCs de tamanho 16×16 para quadros I.

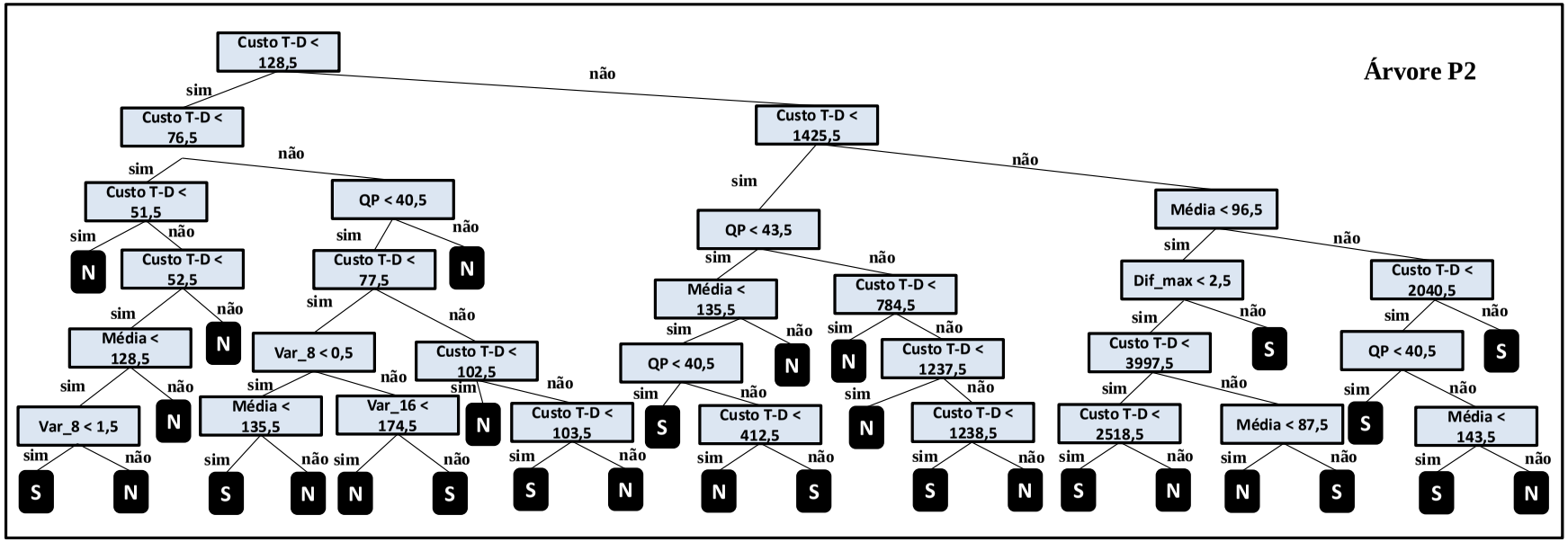


Figura 28 – Ilustração da “Árvore P2” da árvore de decisão para UCs de tamanho 16×16 para quadros I.

Apêndice B – Árvores de decisão para UCs 32×32 e 16×16 em quadros P/B

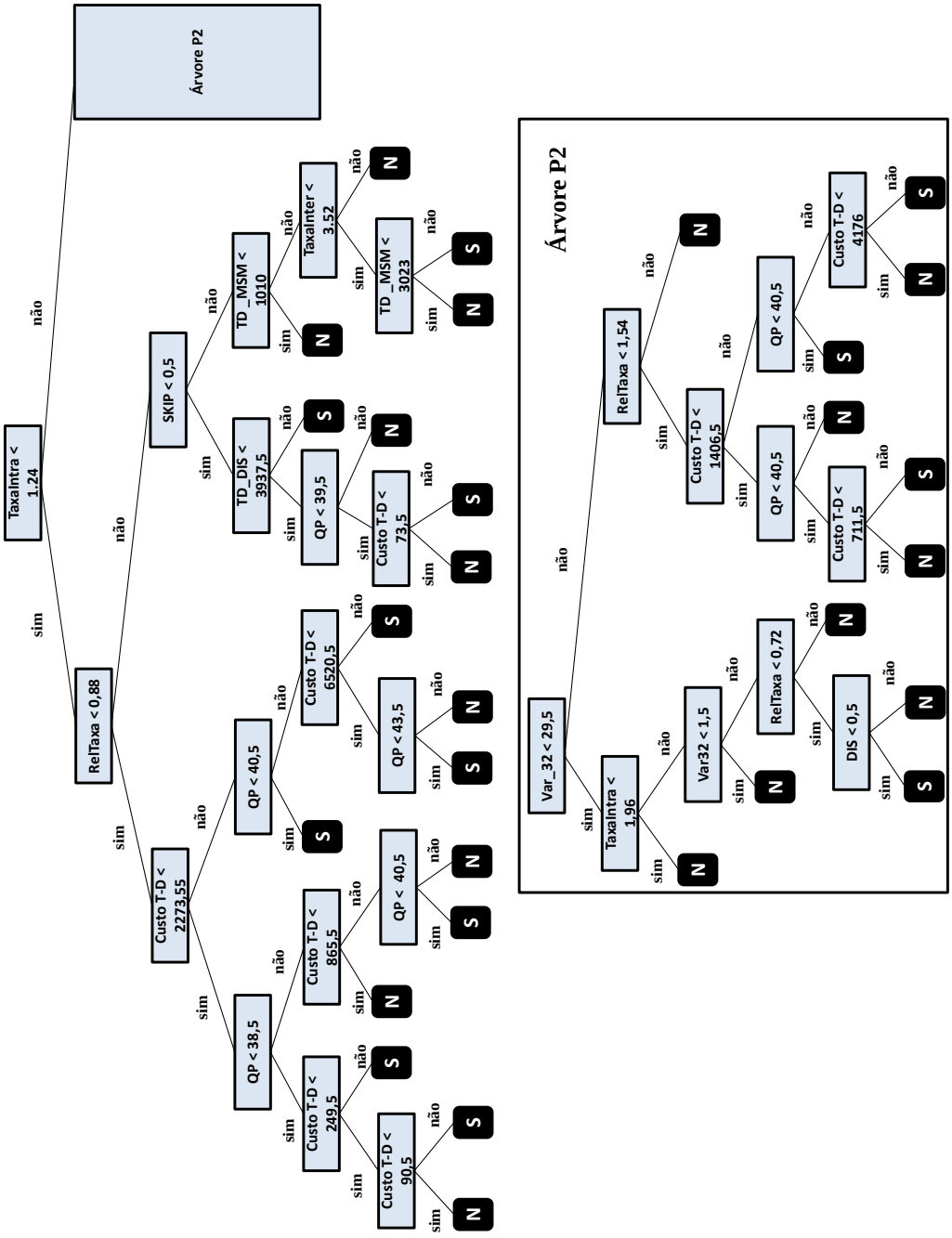


Figura 29 – Ilustração da árvore de decisão para UCs de tamanho 32×32 para quadros P e B.

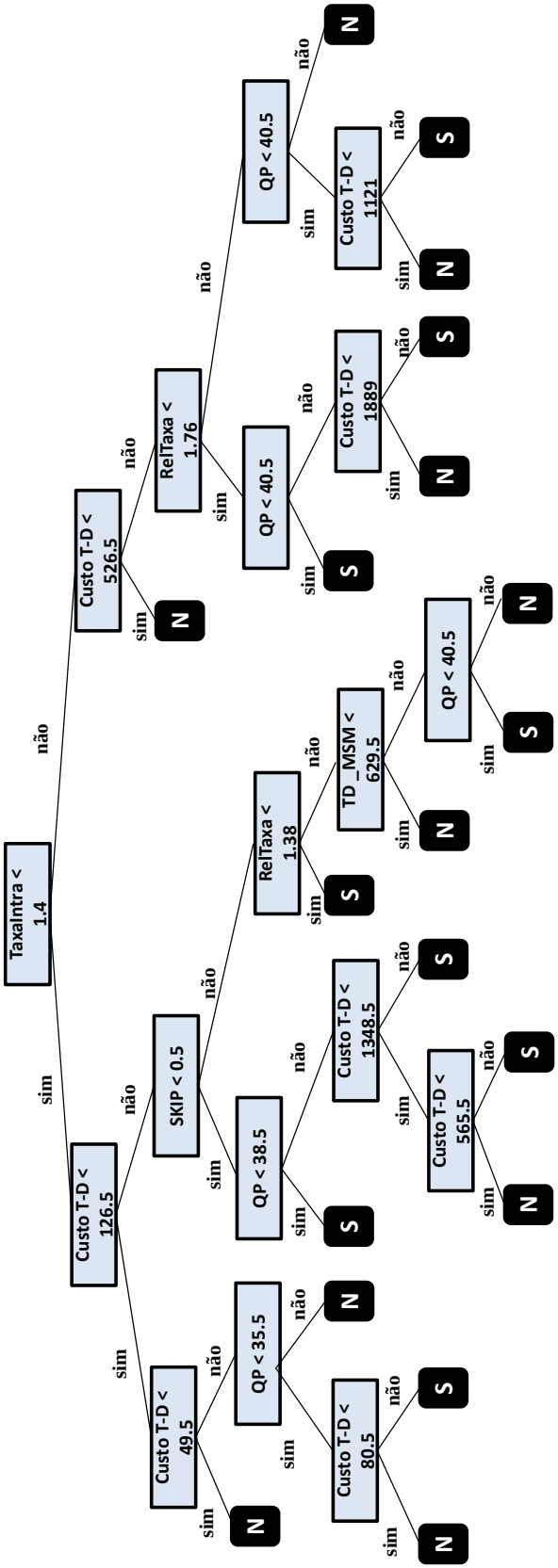


Figura 30 – Ilustração da árvore de decisão para UCs de tamanho 16x16 para quadros P e B.

Apêndice C – Sequências de teste

Este apêndice detalha as características das sequências de testes utilizadas para as avaliações neste trabalho.

Primeiramente são apresentadas as sequências de resolução 1024×768 pixels: *Balloons*, *Kendo* e *Newspaper1*. As três sequências foram capturadas com uma taxa de amostragem de 30 quadros por segundo.

A Figura 31 apresenta a imagem de textura para a sequência *Balloons* e a Figura 32 apresenta seu mapa de profundidade associado. Esta sequência é caracterizada por ter sido gravada em um ambiente interno, feita em estúdio, com movimentação de objetos complexa (com reflexões e transparências), movimentação da câmera, nível médio de detalhes, estrutura de profundidade com complexidade média e luz de estúdio.

Figura 31 – Imagem de textura da sequência *Balloons*.

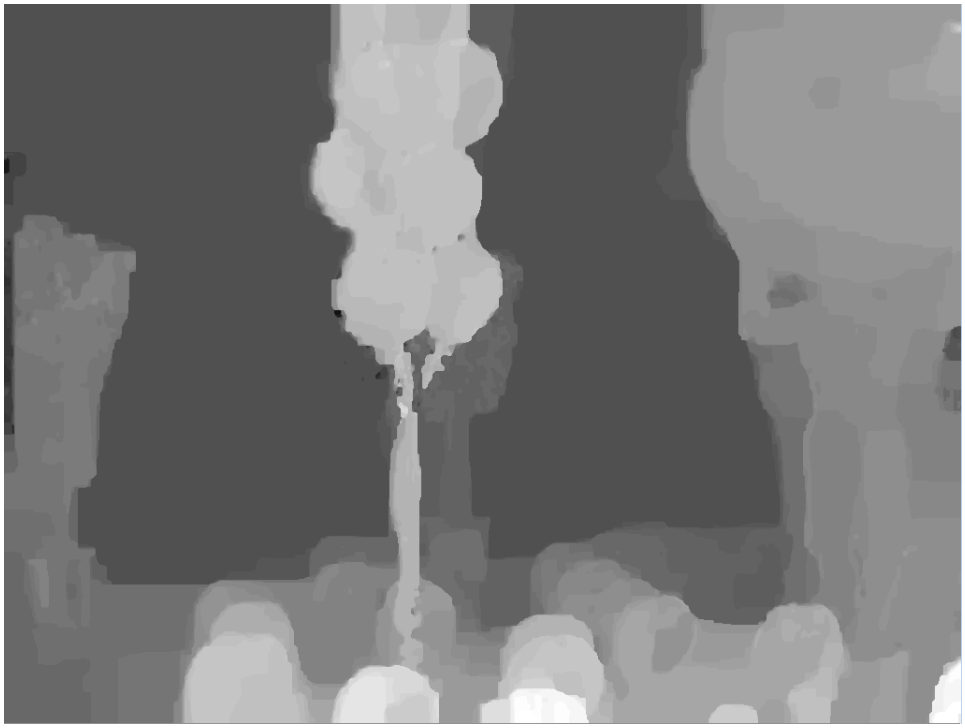


Figura 32 – Imagem do mapa de profundidade da sequência *Balloons*.

A sequência *Kendo* também é caracterizada por ter sido capturada em ambiente interno, feita em estúdio, com movimentação complexa de objetos (incluindo fumaça, reflexões e transparência), movimentação da câmera, alto nível de detalhes, estrutura de profundidade com complexidade média e luz de estúdio. Na Figura 33 e na Figura 34 são exibidos as imagens de textura e o mapa de profundidade para a sequência *Kendo*, respectivamente.

Figura 33 – Imagem de textura da sequência *Kendo.*

Figura 34 – Imagem do mapa de profundidade da sequência *Kendo*.

As características da sequência *Newspaper1* são: ambiente interno, feita em estúdio, movimentação simples dos objetos, câmera estática, alto nível de detalhes, estrutura de profundidade com complexidade média e luz de estúdio. A Figura 35 e a Figura 36 apresentam uma imagem da sequência *Newspaper1* e seu mapa de profundidade, respectivamente.

Figura 35 – Imagem de textura da sequência *Newspaper1*.

Figura 36 – Imagem do mapa de profundidade da sequência *Newspaper1*.

As sequências *GT_Fly*, *Poznan_hall2*, *Poznan_Street*, *Undo_Dancer* e *Shark* possuem resolução de 1920×1088 pixels e foram filmadas a uma taxa de amostragem de 25 quadros por segundo, exceto a sequência *Shark* que foi filmada com uma taxa de 30 quadros por segundo.

A sequência *GT_Fly* é uma sequência de vídeo sintética que foi criada utilizando recursos de computação gráfica e apresenta as seguintes características: movimentação simples de objetos, movimentação da câmera, nível médio de detalhes e estrutura de profundidade complexa. A Figura 37 e a Figura 38 mostram um quadro da sequência *GT_Fly* e seu mapa de profundidade, respectivamente.

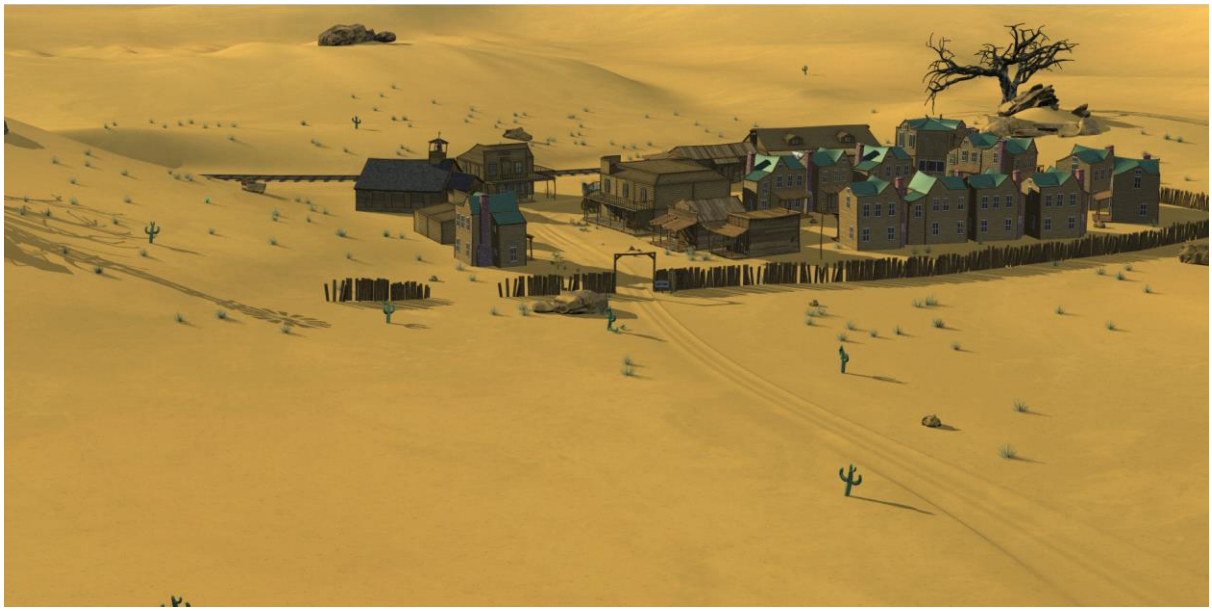


Figura 37 – Imagem de textura da sequência *GT_Fly*.

Figura 38 – Imagem do mapa de profundidade da sequência *GT_Fly*.

A sequência *Poznan_Hall2* foi filmada dentro da *Poznan University of Technology.* Esta sequência caracteriza-se por ser filmada em ambiente interno, movimentação de objetos complexas (incluindo reflexões e transparência), movimentação da câmera, nível médio de detalhes e uma estrutura de profundidade complexa. Na Figura 39 é apresentada uma imagem de textura da sequência e seu mapa de profundidade na Figura 40.

Figura 39 – Imagem de textura da sequência *Poznan_Hall2*.

Figura 40 – Imagem do mapa de profundidade da sequência *Poznan_Hall2*.

A sequência *Poznan_Street* é apresentada na Figura 41 e seu mapa de profundidade na Figura 42. Esta sequência é caracterizada por ser gravada em um ambiente externo, possuir movimentação complexas de objetos, câmera sem movimentação, alto nível de detalhes, estrutura de profundidade complexa e utilização de luz natural.

Figura 41 – Imagem de textura da sequência *Poznan_Street*.

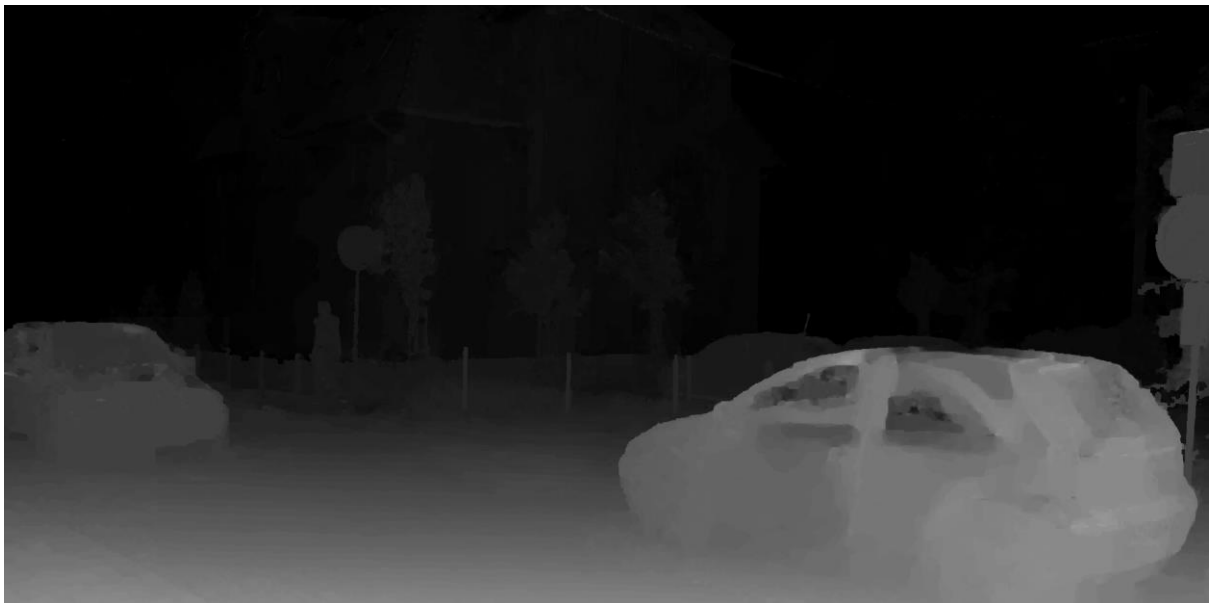


Figura 42 – Imagem do mapa de profundidade da sequência *Poznan_Street*.

Assim como a sequência *GT_Fly*, a sequência *Undo_Dancer* é um vídeo sintético gerado utilizando recursos de computação gráfica. Esta sequência possui movimentação complexa de objetos, câmera com movimentação, alto nível de detalhes e a complexidade da estrutura de profundidade é simples. A Figura 43 e a Figura 44 apresentam um quadro da sequência e seu mapa de profundidade, respectivamente.

Figura 43 – Imagem de textura da sequência *Undo_Dancer*.

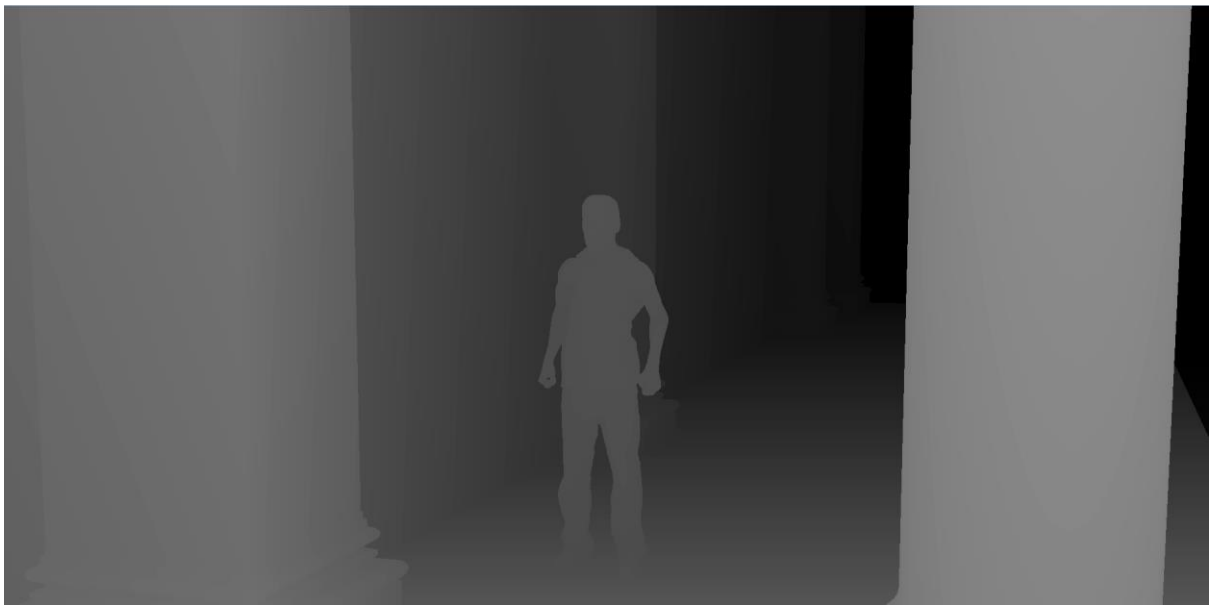


Figura 44 – Imagem do mapa de profundidade da sequência *Undo_Dancer*.

A sequência *Shark* também é um vídeo sintético que foi gerada através de recursos de computação gráfica com movimentação complexa de objetos, com movimentação da câmera, alto nível de detalhes e estrutura de profundidade complexa. A imagem de textura e seu mapa de profundidade associados são apresentados na Figura 45 e na Figura 46, respectivamente.

Figura 45 – Imagem de textura da sequência *Shark*.

Figura 46 – Imagem do mapa de profundidade da sequência *Shark*.

A Tabela 10 sumariza as características das sequências apresentadas. Estas características foram obtidas em (MOBILE-3DTV, 2013) e quando indisponíveis foram inferidas através de análises após a visualização das sequências de vídeos.

Tabela 10 – Características das sequências de teste.

Resolução	Vídeo	FPS	Ambiente	Movimentação dos objetos	Câmera	Nível de detalhes	Estrutura da profundidade
1024×768	Balloons	30	Interno	Complexa	Móvel	Médio	Média
	Kendo	30	Interno	Complexa	Móvel	Alto	Média
	Newspaper1	30	Interno	Simples	Estática	Alto	Média
1920×1088	GT_Fly	25	Comp. Gráfica	Simples	Móvel	Médio	Complexa
	Poznan_Hall2	25	Interno	Complexa	Móvel	Médio	Complexa
	Poznan_Street	25	Externo	Complexa	Estática	Alto	Complexa
	Undo_Dancer	25	Comp. Gráfica	Simples	Móvel	Alto	Simples
	Shark	30	Comp. Gráfica	Complexa	Móvel	Alto	Complexa

Apêndice D – Lista das principais publicações durante o mestrado

1. Título: “Energy-aware scheme for the 3D-HEVC depth maps prediction”

- Journal: Journal of Real-Time Image Processing (JRTIP), 2016. (Qualis B1)

2. Título: “Block-level fast coding scheme for depth maps in three-dimensional high efficiency video coding”

- Journal: SPIE Journal of Electronic Imaging (JEI), 2018. (Qualis A2)

3. Título: “Low-Power and High-Throughput Hardware Design for the 3D-HEVC Depth Intra Skip”

- Evento: IEEE International Symposium on Circuits & Systems (ISCAS), 2017. (Qualis A1)

4. Título: “Edge-Aware Depth Motion Estimation - A Complexity Reduction Scheme for 3D-HEVC”

- Evento: European Signal Processing Conference (EUSIPCO), 2017. (Qualis A2)

5. Título: “Depth Modeling Modes Complexity Control System for the 3D-HEVC Video Encoder”

- Evento: European Signal Processing Conference (EUSIPCO), 2017. (Qualis A2)

6. Título: “Real-time simplified edge detector architecture for 3D-HEVC depth maps coding”

- Evento: IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2016. (Qualis B1)

Aceitos para publicação:

7. Título: “FAST 3D-HEVC DEPTH MAPS INTRA-FRAME PREDICTION USING DATA MINING”

- Evento: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018. (Qualis A1)

Em desenvolvimento:

8. Título: “Aceleração da Codificação de Mapas de Profundidade no Padrão 3D-HEVC de Codificação de Vídeos Através de Árvores de Decisão Construídas com Mineração de Dados e Aprendizado de Máquina”

- Patente em fase final de análise no Escritório de Transferência de Tecnologia (ETT) na PUCRS.