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

**Dissertação**

# DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC

**Gustavo Freitas Sanchez**

**Pelotas, 2014**

**Gustavo Freitas Sanchez**

# DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC

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

Co-Orientador: Prof. Dr. Marcelo Schiavon Porto

Pelotas, 2014

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S211d Sanchez, Gustavo Freitas

DMMFast : esquema de redução de complexidade para a predição intra de mapas de profundidade no padrão emergente 3D-HEVC / Gustavo Freitas Sanchez ; Luciano Volcan Agostini, orientador ; Marcelo Schiavon Porto, coorientador. — Pelotas, 2014.

93 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2014.

1. Codificação de vídeo 3D. 2. Redução de complexidade. 3. Mapas de profundidade. 4. DMMS. I. Agostini, Luciano Volcan, orient. II. Porto, Marcelo Schiavon, coorient. III. Título.

CDD : 005

Elaborada por Aline Herbstrith Batista CRB: 10/1737

**Gustavo Freitas Sanchez**

**DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC**

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

**Data da Defesa: 29 de setembro de 2014**

**Banca examinadora:**

---

**Prof. Dr. Luciano Volcan Agostini (Orientador)**
**Doutor em Computação pela Universidade Federal do Rio Grande do Sul**

---

**Prof. Dr. Marcelo Schiavon Porto (Co-Orientador)**
**Doutor em Computação pela Universidade Federal do Rio Grande do Sul**

---

**Prof. Dr. Rafael Iankowski Soares**
**Doutor em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul**

---

**Prof. Dr. Júlio Carlos Balzano de Mattos**
**Doutor em Computação pela Universidade Federal do Rio Grande do Sul**

---

**Prof.ª Dr.ª Carla Liberal Pagliari**
**Doutor em Electronic Systems Engineering pela *University of Essex***

## AGRADECIMENTOS

Primeiramente agradeço a Deus por essa conquista. Agradeço a tudo que ele me proporcionou nesses anos de vida e a possibilidade de eu fazer o trabalho desta dissertação.

Também agradeço aos meus pais Luiz Antônio Sanchez e Maria Helena Sanchez. A eles agradeço a vida que me proporcionaram ao longo destes 23 anos, aos conhecimentos que me transmitiram e toda a ajuda que me deram em toda a minha vida.

Gostaria de agradecer aos meus amigos e orientadores Luciano Agostini, Marcelo Porto e Bruno Zatt por me permitirem fazer parte deste grupo, por me auxiliarem a obter melhores resultados e explicações no meu trabalho e por acreditarem que meu trabalho poderia render bons resultados mesmo em pouco tempo de mestrado. Além disso, me permitiram ter uma nova experiência nessa fase do mestrado: a de gestão de pessoas, permitindo-me ter discricionariedade para selecionar as atividades de dois bolsistas que ficariam sobre meu comando. Esses dois bolsistas, Mário Saldanha e Gabriel Balota, me auxiliaram em praticamente todas as atividades de pesquisa do meu mestrado e lhes agradeço por isso.

Agradeço todos os meus colegas e professores do laboratório GACI por fazerem o ambiente de trabalho ser agradável e competitivo.

Existem amigos que fazem parte de apenas alguns momentos da nossa vida, mas poucos são para a vida inteira. Dessa forma, agradeço aos meus “irmãos” Joaquim, Igor, Rodrigo, Alexandre, Eduardo, Pedro e Alessandro.

Finalmente, agradeço minha namorada Daniele por fazer parte da minha vida, me incentivar a ser o melhor e me compreender em todos os momentos.

**Obrigado a todos!**

***“No meio da dificuldade encontra-se a oportunidade”***

*- ALBERT EINSTEIN*

## Resumo

Sanchez, Gustavo Freitas. **DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC.** 2014. 93f. Dissertação (Mestrado) – Programa de Pós Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas.

Esta dissertação apresenta um esquema de redução de complexidade para a predição intra-quadro de mapas de profundidade no padrão 3D *High Efficiency Video Coding* (3D-HEVC), chamada DMMFast (*Depth Modeling Modes Fast Prediction*). O 3D-HEVC introduziu um novo conjunto de ferramentas na codificação de mapas de profundidade (vistas que indicam a distância entre os objetos e a câmera), ocasionando uma elevação na complexidade computacional. Esse aumento resultou em novos desafios de pesquisa para a redução de complexidade. O DMMFast é um esquema composto de dois algoritmos, também desenvolvidos neste trabalho: o *Simplified Edge Detector* (SED) e o *Gradient-Based Mode One Filter* (GMOF). O SED classifica os blocos que devem ser codificados utilizando apenas os modos intra tradicionais, removendo avaliações desnecessárias dos novos modos, chamados de *Depth Modeling Modes* (DMM), que apresentam elevado custo computacional. O GMOF visa reduzir a complexidade do DMM 1, o mais complexo dentre os novos modos de predição, aplicando um filtro de gradientes nas bordas do bloco e determinando as posições mais promissoras para serem avaliadas pelo DMM 1. Avaliações em software demonstram que o DMMFast é capaz de atingir uma redução de complexidade (tempo de codificação) de 24,5% na predição de mapas de profundidade, sem afetar a qualidade das vistas sintetizadas. Considerando o cenário *All Intra*, o DMMFast é capaz de atingir, em média, uma redução de complexidade de 64% na codificação dos mapas de profundidade. Uma avaliação subjetiva de qualidade também foi realizada cujos resultados demonstram que a técnica proposta insere perdas de qualidades desprezíveis no vídeo codificado.

**Palavras-chave**: codificação de vídeo 3D; redução de complexidade; mapas de profundidade; DMMs.

## Abstract

Sanchez, Gustavo Freitas. **DMMFast: Esquema de Redução de Complexidade para a Predição Intra de Mapas de Profundidade no Padrão Emergente 3D-HEVC.** 2014. 93f. Dissertação (Mestrado) – Programa de Pós Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas.

This work presents a complexity reduction scheme for the depth maps intra-frame prediction of the 3D High Efficiency Video Coding (3D-HEVC), called DMMFast (Depth Modeling Modes Fast Prediction). The 3D-HEVC introduces a new set of specific tools for depth map coding, inserting additional complexity to intra prediction, which increases the computational complexity. This increase in complexity results in new challenges in terms of complexity reduction. Therefore, this work presents the DMMFast, a scheme composed of two new algorithms: the Simplified Edge Detector (SED) and the Gradient-Based Mode One Filter (GMOF). The SED classifies the blocks that are likely to be better predicted by the traditional intra modes, avoiding the evaluation of the new modes, called Depth Modeling Modes (DMM), which represents a high computational complexity in the 3D coding. The GMOF is focused on reducing the DMM 1 complexity, the most complexity mode inserted in 3D-HEVC, applying a gradient-based filter in the borders of the block and predicts the best positions to evaluate the DMM 1. Software evaluations showed that DMMFast is capable to achieve a time saving of 24.5% on depth maps prediction, without affecting the quality of the synthesized views. Considering the All Intra setting, the DMMFast is capable to achieve, in average, a time saving of 64% for depth maps coding. Subjective quality assessment was also performed, showing that the proposed technique inserts minimum quality losses in the final encoded video.



**Key-words**: 3D video coding, complexity reduction, depth maps, DMMs.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ADLF | *Availability Deblocking Loopback Filter* |
| AI | *All-Intra* |
| BD-Rate | *Bjøntegaard Delta Rate* |
| CABAC | Codificação Aritmética Binária Adaptativa ao Contexto |
| CCT | Condições Comuns de Teste |
| CPV | *Constant Partition Value* |
| CTU | *Coding Tree Unit* |
| CU | *Coding Unit* |
| DCP | *Disparity Compensated Prediction* |
| DCR | *Degradation Category Rating* |
| DCT | *Discrete Cosine Transform* |
| DF | *Deblocking Filter* |
| DGLF | *Depth-Gradient-based Loopback Filterer* |
| DMM | *Depth Modeling Mode* |
| DMMFast | *Depth Modeling Modes Fast Prediction* |
| DIBR | *Depth-image-based rendering* |
| DLT | *Depth Lookup Table* |
| FVV | *Free View point Video* |
| GFS | *Greedy and Fleeting Step* |
| GIST | *Gwangju Institute of Science and Technology* |
| GMOF | *Gradient-based Mode One Filter* |
| GOP | *Group of Pictures* |
| HD | *High Definition* |
| HEVC | *High Efficiency Video Coding* |
| HTM | *3D-HEVC Test Model* |
| IC | Compensação de Iluminação |
| JCT-3V | *Joint Collaborative Team on 3D Video Coding Extension Development* |
| MCP | *Motion Compensated Prediction* |
| MPM | *Most Probable Mode* |
| MVD | *Multi-View Plus Depth* |
| PSNR | *Peak Signal-to-Noise Ratio* |

| | |
|---|---|
| PU | *Prediction Unit* |
| QP | *Quantization Parameter* |
| RA | *Random Access* |
| RD | *Rate-Distortion* |
| RDO | *Rate-Distortion Optimization* |
| RMD | *Rough Mode Decision* |
| SAO | *Sample Adaptive Offset* |
| SDC | *Simplified Depth Coding* |
| SED | *Simplified Edge Detector* |
| TH | *Threshold* |
| TU | *Transform Unit* |
| VSRS | *View Synthesis Reference Software* |
| ZZC | $Z_{far}$ $Z_{near}$ *Compensated Weight Prediction* |
| 2D | *Two-Dimensional* |
| 3D | *Three-Dimensional* |
| 3D-HEVC | 3D-*High Efficiency Video Coding* |
| 3DVC-HTM | 3DVC-HEVC *Test Model* |

**SUMÁRIO**

# 1 INTRODUÇÃO

As aplicações 3D estão ganhando espaço nos últimos anos. Isto está ocorrendo porque o custo de equipamentos que gravam e reproduzem vídeos 3D está diminuindo e o interesse em pesquisa por desenvolvimento de novas tecnologias mais eficientes para codificar vídeos 3D aumentando. Essa popularização ocorre porque vídeos 3D trazem uma nova experiência visual, que transmite ao usuário uma percepção de profundidade da cena. Isso permite que os usuários desfrutem uma experiência mais agradável do que as apresentadas nos vídeos 2D (YASAKETHU, 2009). Este cenário motivou vários pesquisadores a investigar e desenvolver novas soluções para novas aplicações de vídeos 3D tais como televisão 3D e *Free View point Video* (FVV) (TANIMOTO, 2009).

Um vídeo 3D é filmado por várias câmeras simultaneamente. Uma vista é o conjunto de quadros gravados por uma das câmeras (um quadro para cada instante de tempo), também chamado de vista (ou canal) de textura.

Um codificador 3D deve ser capaz de receber dados de diferentes câmeras (vistas) e fazer a codificação de forma que os dados no *bitstream* possam ser decodificados por um decodificador 2D do mesmo padrão (apenas a vista base), um decodificador de vistas estéreo e/ou um decodificador de vistas 3D (CHEN, 2008), como é mostrado na Figura 1.

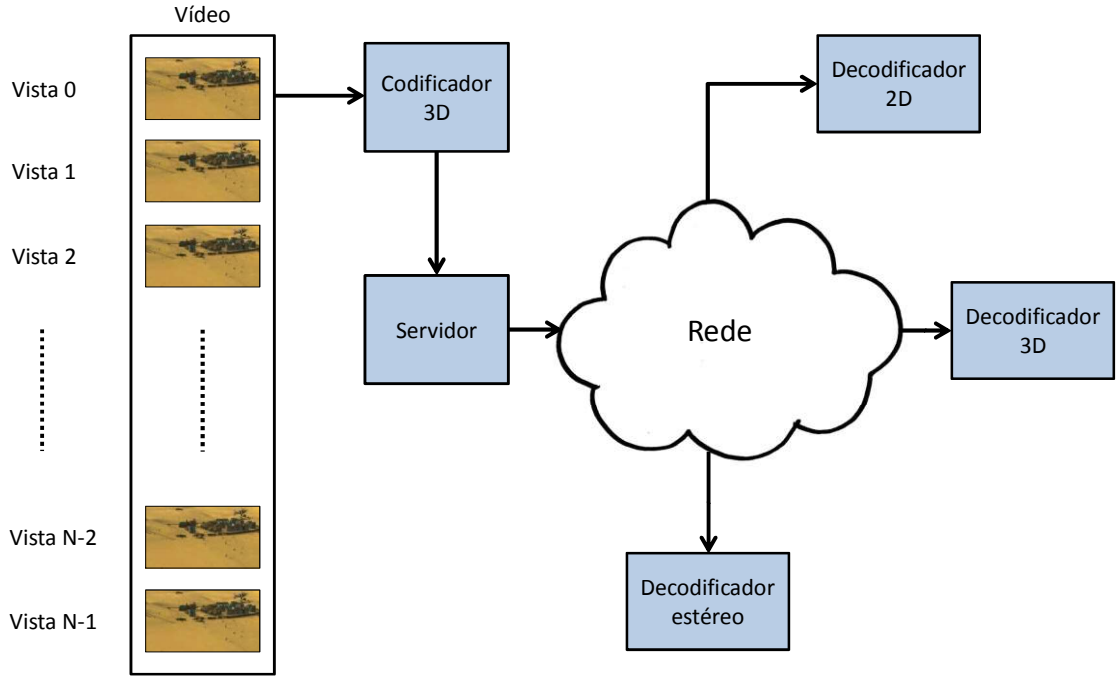



**Figura 1 – Sistema de codificação multi-vistas.**

A codificação de vídeos 2D visa reduzir as informações presentes nos vídeos, codificando características redundantes entre quadros de instantes diferentes de tempo (redundância temporal), dentro de um próprio quadro (redundância espacial e/ou na codificação binária do vídeo (redundância entrópica). As redundâncias de quadros em instantes de tempo diferentes são codificadas utilizando a predição temporal, as redundâncias dentro de um próprio quadro são codificadas pela predição intra-quadro (foco deste trabalho) e a codificação entrópica é responsável pela codificação da representação binária do vídeo (RICHARDSON, 2010).

Todas essas ferramentas são utilizadas na codificação 3D e, além disso, são utilizadas ferramentas para reduzir informações redundantes entre as diferentes vistas, o que é feito pela predição inter-vistas (também chamada de disparidade).

O modelo *Multi-View Plus Depth* (MVD) é um novo formato utilizado na codificação de vídeos 3D, onde cada canal de textura está associado a um canal de profundidade. Por um lado, quadros de textura, que são as imagens naturais e coloridas que vemos na TV, são compostas por informações de luminância e crominâncias. Por outro lado, os mapas de profundidade são representados apenas por informações de luminância, o que os torna imagens em tons de cinza. Estes canais de profundidade são representados por imagens em tons de cinza, chamadas de mapas de profundidade, que indicam a distância entre a câmera e o objeto capturado.

A vantagem no uso de canais de profundidade associados a canais de textura é que é possível gerar os vídeos de canais intermediários (virtuais) de forma eficiente através da interpolação de alguns canais de textura e profundidade. Isto reduz drasticamente a necessidade de processamento para codificação de vídeo e armazenamento e/ou largura de banda para transmissão do vídeo, pois reduz o número de vistas a serem transmitidas (MERKLE, 2007a). Isso pode ser notado analisando a Figura 2, onde em um vídeo com nove câmeras com seus respectivos canais de profundidade associados são apresentados. É possível codificar as câmeras um, cinco e nove, com os seus canais de profundidade associados. Com isso os outros canais de textura desejados podem ser sintetizados no decodificador, reduzindo a quantidade de informações codificadas e transmitidas. Neste caso, as vistas 2, 3, 4, 6, 7 e 8 são sintetizadas durante a apresentação das cenas, no decodificador.

Analisando um vídeo 2D sem compressão, com duração de 10 minutos, resolução HD 1080p (1920x1080 pixels), com amostras de oito bits, capturado a uma taxa de 30 quadros por segundo e com uma taxa de subamostragem de 4:2:0, é necessário aproximadamente 56 GB para armazená-lo e, além disso, uma largura de banda de 0,75 Gbps para transmiti-lo em tempo real. Um vídeo 3D de cinco vistas de textura sem codificação, e com as mesmas características, necessita de 280 GB para ser armazenado e a largura de banda de 3,73 Gbps para a sua transmissão em tempo real. Se adicionarmos canais de profundidade, o volume de dados cresce em 66,67%, uma vez que o canal de profundidade possui apenas dados de luminância, enquanto o canal de textura possui dados de luminância e crominância. Isso demonstra claramente que é inviável armazenar ou transmitir vídeos sem compressão, principalmente para vídeos de alta definição, como HD 1080p, sendo que o problema se torna ainda maior quando consideramos vídeos 3D.


**Figura 2 – Canais de textura e profundidade transmitidos e os canais virtuais gerados.**

## 1.1 Motivação

Vídeos sem codificação ocupam um espaço muito grande de memória para armazenamento e uma largura de banda inviável para transmissão em tempo real. O 3D-*High Efficiency Video Coding* (3D-HEVC) (TECH, 2013a) (TECH, 2013b) (MULLER, 2013) é o padrão emergente de codificação de vídeos 3D. Ele consegue obter altas taxas de compressão e qualidade de vídeo. Ele é baseado no padrão de codificação de vídeo *High Efficiency Video Coding* (HEVC) (JCT-VC, 2013). Como o padrão emergente 3D-HEVC ainda está em fase de desenvolvimento, existe a possibilidade de incorporação de diversos algoritmos inovadores para aumentar o desempenho da codificação. Por ser uma ferramenta nova para a codificação de vídeos 3D, existem poucos algoritmos desenvolvidos para a codificação do canal de profundidade. Além disso, os algoritmos aplicados atualmente preveem poucos mecanismos para sua simplificação, o que ocasiona um grande esforço

computacional para fazer a codificação destes mapas de profundidade. Com isto, a principal motivação para este trabalho é ser um dos primeiros trabalhos a explorar a redução de complexidade dos modos de predição intra-quadros, utilizados na codificação de mapas de profundidade do padrão 3D-HEVC.

## 1.2 Objetivo

O objetivo desse trabalho é desenvolver um esquema de redução de complexidade para a predição intra-quadro em mapas de profundidade. Este esquema foi chamado de *Depth Modeling Modes Fast Prediction* (DMMFast). Para avaliar a qualidade dos algoritmos resultante desta simplificação, o DMMFast foi avaliado no software de referência do 3D-HEVC: o 3D-HEVC *Test Model* (HTM) (3D-HTM, 2013). Em todo este trabalho será considerada a versão 7.0 do HTM.

## 1.3 Organização do trabalho

Este trabalho divide-se em sete capítulos, de forma a apresentar os principais conceitos da codificação de vídeo 3D presentes em um codificador do padrão emergente 3D-HEVC, além dos trabalhos desenvolvidos. O segundo capítulo apresenta brevemente a estrutura de codificação do codificador HEVC focando em explicar principalmente as características da predição intra de componentes de textura. Faz-se oportuno relembrar que o padrão emergente 3D-HEVC utiliza o padrão HEVC como base, apenas adicionando novas ferramentas. No terceiro capítulo é apresentado especificamente o padrão emergente 3D-HEVC, abordando as novidades presentes nesse padrão. No capítulo três também são apresentadas, detalhadamente, explicações sobre mapas de profundidade e os algoritmos inovadores que o 3D-HEVC apresenta para sua codificação. No quarto capítulo são apresentados os principais desafios da codificação de vídeo 3D com mapas de profundidade associados, além de uma revisão bibliográfica de trabalhos propostos na literatura focando a codificação de vídeo 3D. Ainda neste capítulo, são mostradas as principais limitações destes trabalhos que motivaram a criação do DMMFast. O quinto capítulo apresenta o referencial teórico sobre os algoritmos propostos para a redução da complexidade através de uma análise qualitativa dos mapas de profundidade. Além disso, o DMMFast também é apresentado neste capítulo. No sexto capítulo são especificadas as Condições Comuns de Teste (CCT), usadas para as comparações entre algoritmos para a codificação de vídeos 3D, e também são apresentados os resultados das análises de software, uma análise subjetiva de

qualidade e comparação com trabalhos relacionados. Finalmente, o sétimo capítulo finaliza o trabalho, apresentando as conclusões e trabalhos futuros sobre o tema abordado.

# 2 O PADRÃO HEVC

Este capítulo tem por objetivo apresentar os métodos de codificação utilizados pelo padrão HEVC, que são utilizados como base para o padrão emergente 3D-HEVC. O principal foco deste trabalho é a predição intra-quadro de mapas de profundidade e, os algoritmos aplicados na predição intra-quadro de textura são aplicados integralmente na codificação de mapas de profundidade. Por este motivo, faz-se necessário descrevê-los detalhadamente ao final deste capítulo.

## 2.1 Fluxo de codificação do HEVC

Os vídeos digitais são formados por várias imagens (também chamadas de quadros) que são capturados em um curto espaço de tempo. Este curto espaço de tempo necessita ser de pelo menos 24 quadros por segundo a fim de transmitir a sensação de movimento ao espectador.

O padrão HEVC adota um esquema de codificação quadro a quadro. Para simplificar a codificação e obter uma maior eficiência, cada um destes quadros é dividido em blocos menores, e cada um destes blocos passa individualmente por um processo completo de codificação. O diagrama de blocos do codificador HEVC é apresentado na Figura 3.

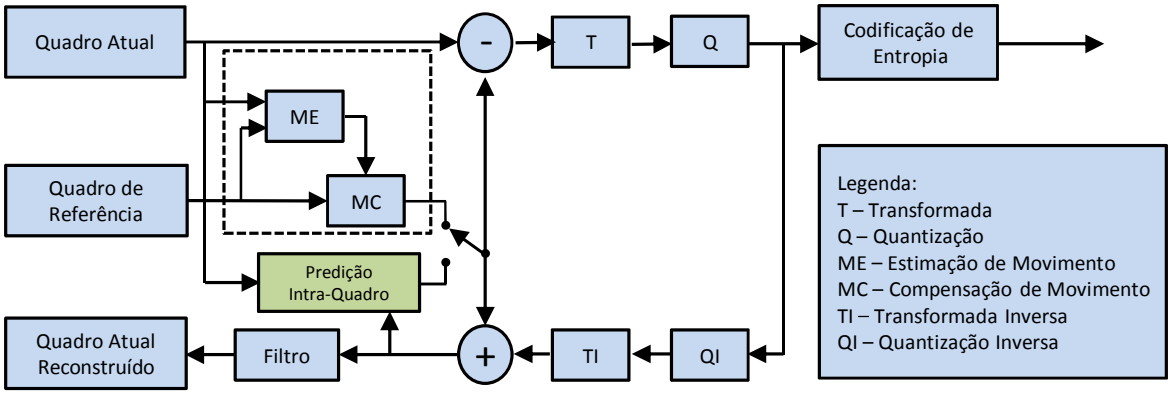



**Figura 3 – Diagrama de blocos do codificador *HEVC*.**

O padrão HEVC define que cada quadro é dividido em *treeblocks* ou *Coding Tree Units* (CTU). Cada CTU está limitada a um tamanho máximo de 64x64 amostras, e o tamanho da CTU deve ser definido antes do inicio da codificação, sem possibilidade de alterá-lo dinamicamente (SULLIVAN, 2012).

Cada CTU pode ser dividida em Unidades de Codificação (*Coding Units* – CU), onde é possível selecionar o modo de predição (inter-quadro ou intra-quadro) a

ser utilizado dentro daquela CU. Essa divisão é feita através do particionamento da CTU em quatro blocos de tamanhos iguais. Cada uma dessas CUs pode ser dividida novamente em quatro blocos menores, e este processo de divisão pode ser repetido até que o tamanho mínimo de 8x8 amostras seja obtido (SULLIVAN, 2012).

A Figura 4 mostra esse processo de divisão onde é formada uma árvore quaternária (*Quadtree*) (SILVA, 2013). Nesta figura a CTU tem um tamanho de 64x64 amostras. As folhas desta *Quadtree* representam o tamanho da CU escolhida, onde o menor nível (8x8) ocorre na profundidade 4.

Assim, cada CU é subdividida em Unidades de Predição (*Prediction Units* – PU), armazenando as informações importantes para a predição tal como o tipo de predição, vetores de movimento, entre outros. As PUs podem ser representadas em tamanhos simétricos e assimétricos. A Figura 5 mostra todas as possíveis representações das PUs simétricas e assimétricas (JCT-VC, 2013).

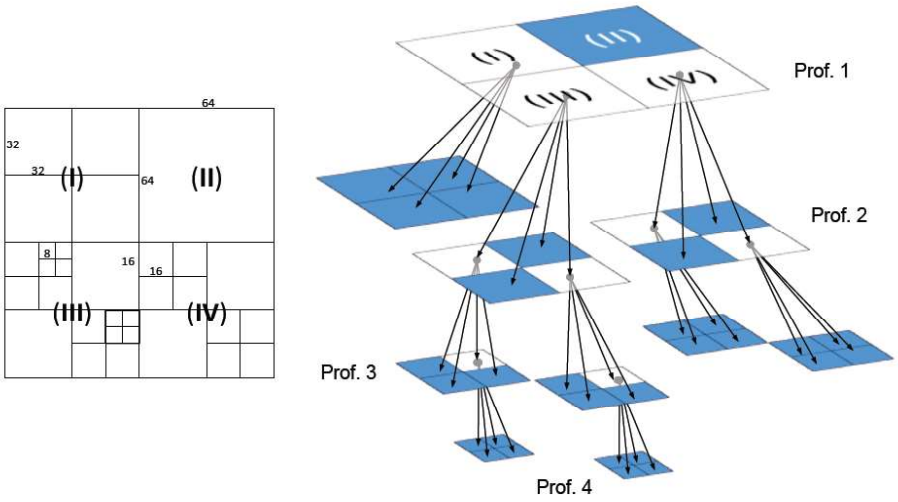



**Figura 4 – *Quadtree* das divisões de CTUs (SILVA, 2013).**

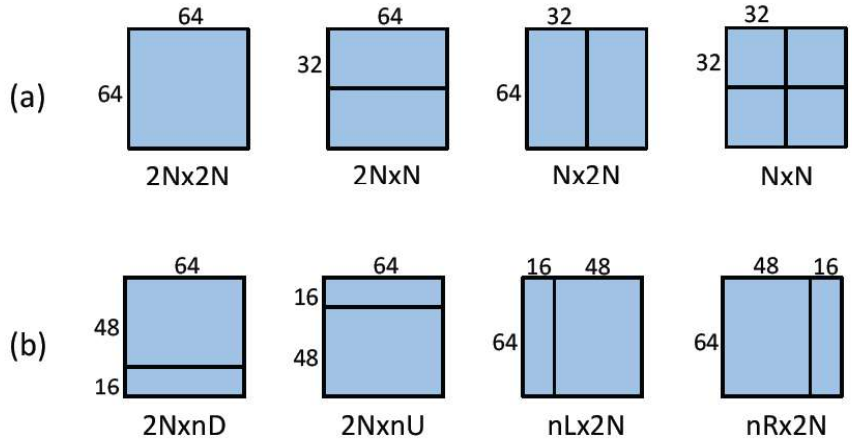



**Figura 5 – Representações (a) simétricas e (b) assimétricas de PU (JCT-VC, 2013).**

### 2.1.1 O processo de predição

O processo de predição é o primeiro processo a ser executado por um codificador HEVC e pode ser de três tipos: I, P ou B. Quadros do tipo I são codificados utilizando apenas a predição intra-quadro, em quadros do tipo P e B podem ser aplicados tanto o processo de predição intra-quadro como o processo de codificação inter-quadros. Nos quadros de tipo B cada CTU pode utilizar vetores de movimento para mais de um quadro de referência na predição inter-quadro, enquanto nos quadros do tipo P é utilizado apenas vetores de movimento para um único quadro como referência.

A predição intra-quadro é responsável por remover as informações redundantes presentes no quadro, utilizando apenas as informações do próprio quadro. O HEVC define 33 modos direcionais, além dos modos DC e planar para a predição intra-quadro (JCT-VC, 2013). O número total de direções depende do tamanho das PUs em questão: para PUs de tamanho 4x4, apenas 18, dos 35 modos possíveis, podem ser usados; para PUs de tamanho 64x64, apenas 4; já para os demais tamanhos de PUs, todos os 35 tamanhos são possíveis. Além disso, somente partições de tamanho *NxN* e *2Nx2N* são permitidas na predição intra-quadro.

A predição inter-quadro é responsável por encontrar informações redundantes presentes entre quadros temporalmente vizinhos. As informações entre quadros vizinhos são muito semelhantes devido à alta frequência de captura de quadros das câmeras (pelo menos 24 quadros por segundo), o que faz com que a predição inter-quadro seja a responsável pela maior parte dos ganhos de compressão obtidos nos padrões de codificação de vídeo atuais (CHENG, 2009). A predição inter-quadro não é o foco deste trabalho, no entanto, maiores detalhes podem ser encontrados em (BHASKARAN, 1999).

Após o processo de predição, o bloco predito é subtraído do bloco original e as diferenças de predição (resíduos) são enviados para as próximas etapas do codificador (BHASKARAN, 1999).

### 2.1.2 Transformadas e quantização

Após a predição de um bloco, os resíduos são enviados para um processo de transformadas. Cada uma das PUs é subdividida em Unidades de Transformada (*Transform Unit* – TU). Neste processo em que aplicamos a Transformada Discreta

do Cosseno (Discrete Cosine Transform – DCT), as informações dos resíduos no domínio espacial são transformadas para o domínio das frequências (SULLIVAN, 2012). Este processo é realizado com a finalidade separar as frequências que possuem um menor impacto na visão humana e preparar os dados para a etapa de quantização. No caso, as altas frequências são as menos significativas para a visão humana e podem ser cortadas, sem apresentar significativo impacto na qualidade subjetiva do vídeo, que é a qualidade observada pelos telespectadores (BASKARAN, 1999).

O processo de quantização é aplicado logo após as transformadas, aplicando uma operação de corte aos coeficientes obtidos pelo processo de transformadas. A etapa de quantização recebe os resíduos no domínio das frequências e aplica operações irreversíveis, eliminando ou atenuando dados de alta frequência. A quantização está diretamente ligada a um parâmetro de quantização (*Quantization Parameter* - QP) que indica a intensidade de perdas inseridas pela quantização. Quanto maior o QP, maiores serão os cortes nos resíduos, o que tende a maiores perdas na qualidade de vídeo. Por outro lado, valores altos de QP contribuem diretamente para a obtenção de altas taxas de compressão.

### 2.1.3 Codificação de entropia

A etapa de Codificação de Entropia recebe como entrada as informações de resíduos enviados pela etapa de quantização. Após a quantização, os blocos de resíduos apresentam características de matrizes esparsas, o que favorece muito a aplicação de algoritmos de codificação de entropia. Além dos resíduos, nesta etapa as informações laterais, que são os cabeçalhos das estruturas de dados do HEVC, também são codificadas. A etapa de Codificação de Entropia aplica algoritmos de codificação sem perdas (por exemplo, o algoritmo de *Huffman*). As técnicas de codificação utilizadas nesta etapa são fortemente baseadas em análises estatísticas e conseguem representar valores com maior probabilidade de ocorrência com um menor número de bits. A saída desta etapa é o *bitstream* do vídeo codificado.

No HEVC, o algoritmo utilizado para a codificação de entropia utiliza o método de Codificação Aritmética Binária Adaptativa ao Contexto (CABAC) (MARPE, 2003).

### 2.1.4 Reconstrução de quadros de referência

O padrão HEVC ainda possui um laço após a etapa de quantização, onde são aplicadas quantizações inversas e transformadas inversas, além da adição dos

resíduos ao quadro predito, para reconstruir o quadro codificado. Aqui também são aplicados filtros para remover artefatos inseridos pela codificação por blocos. Após a aplicação destes filtros, então o quadro reconstruído é armazenado e poderá ser utilizado como referência para a codificação dos próximos quadros do vídeo. Esta etapa é necessária para que o codificador tenha a mesma referência do que o decodificador ao utilizar um quadro como referência.

No HEVC, os filtros inseridos nesta etapa são o *Sample Adaptive Offset* (SAO) (FU, 2012), que é aplicado em áreas cujas amostras possuem intensidade homogêneas e o *Deblocking Filter* (DF) (NORKIN, 2012) que diminui o efeito de bloco gerado pela codificação por CUs .

## 2.2 Predição Intra-quadro

A Figura 6 apresenta os 33 modos direcionais da predição intra-quadro permitidos pelo padrão HEVC. Na Figura 6 as setas indicam a primeira amostra que será utilizada para gerar a PU predita. Todas as outras amostras da PU predita são copiadas de amostras posicionadas em pontos com o mesmo ângulo de inclinação. Cada número na Figura 6 representa o número em que a codificação intra-quadro será representada.

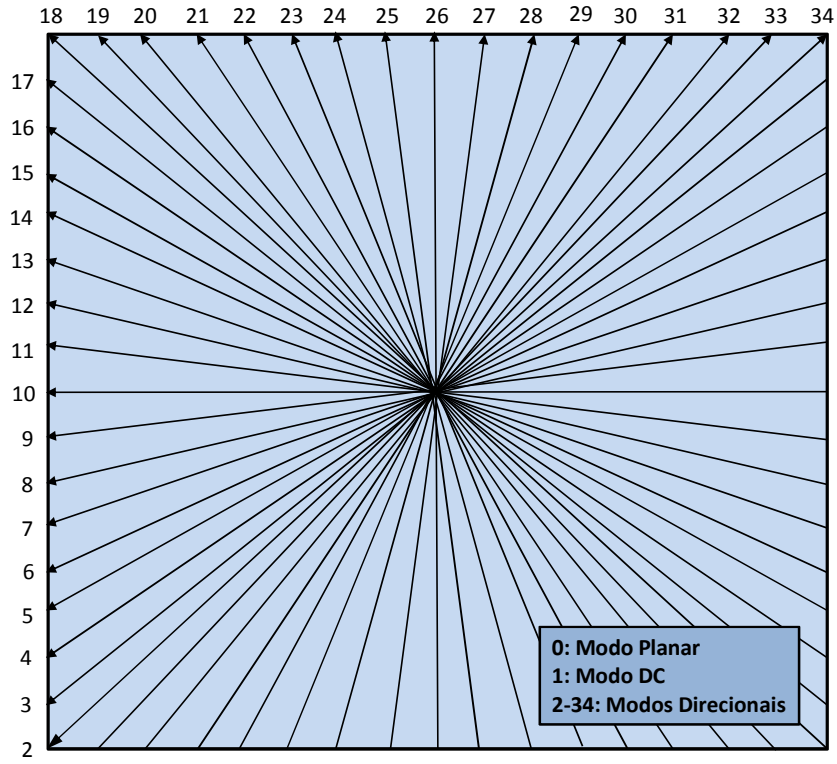



**Figura 6 – Modos de predição intra-quadro possível no HEVC.**

Além destes modos direcionais, existem o modo planar e o modo DC. No modo planar é feito uma interpolação entre as amostras da borda superior e na borda esquerda da PU para determinar as amostras preditas. No modo DC, é calculada a média de todas as amostras da borda superior e esquerda do bloco e então todas as amostras são preditas por este valor médio.

A aplicação de cada um desses modos depende do tamanho do bloco. A Tabela 1 apresenta os modos de predição intra-quadro disponível de acordo com as dimensões da PU.

Todas essas possibilidades de predição intra-quadro fazem com que a decisão sobre qual é o melhor modo de predição seja bastante complexa. O custo computacional desta decisão é elevado, caso o processo de *Full Rate-Distortion Optimization* (RDO) seja realizado. O processo *Full* RDO avalia, para cada bloco, todas as possibilidades de modos de predição, tentando maximizar a qualidade do vídeo e a taxa de compressão. Este custo é elevado e não pode ser tolerado por diversas aplicações.

No software de referência do padrão HEVC é feita uma simplificação deste processo, tal como proposto em (ZHAO, 2011). Primeiramente, todos os modos possíveis são avaliados de forma local, utilizando o critério *Sum of Absolute Transform Differences* (SATD), entre o bloco original e o bloco predito. Após esta etapa, o algoritmo *Rough Mode Decision* (RMD) é responsável por selecionar um número pré-definido de modos e adiciona estes modos em uma lista para ser avaliado pelo custo *Rate-Distortion* (RD) de acordo com o resultado do SATD. A quantidade de modos selecionados pelo algoritmo RMD, em função do tamanho da PU, é apresentada na Tabela 1. Por fim, os *Most Probable Modes* (MPM – Modos mais prováveis) são selecionados para serem avaliados pelo custo RD, de acordo com as informações dos blocos vizinhos, já codificados, acima e a esquerda.

Tabela 1 – Modos de predição intra quadro de acordo com a dimensão da PU.

| Dimensão da PU | Número de modos de predição intra-quadro | Número de Modos Selecionados Utilizando RMD |
|---|---|---|
| 4 | 18 | 8 |
| 8 | 35 | 8 |
| 16 | 35 | 3 |
| 32 | 35 | 3 |
| 64 | 4 | 3 |

Neste capítulo foram apresentadas as principais ferramentas da estrutura de codificação do HEVC. Todas essas ferramentas mencionadas neste capítulo

também são utilizadas na codificação de vídeos 3D, utilizando o padrão emergente 3D-HEVC. Entretanto, o 3D-HEVC não está limitado apenas às ferramentas mencionadas neste capítulo e, no capítulo seguinte, serão apresentadas a estrutura de codificação e as novas ferramentas inseridas nele com foco em vídeos 3D no formato MVD.

# 3 O PADRÃO EMERGENTE 3D-HEVC

Este capítulo tem por objetivo apresentar os conceitos básicos sobre o padrão emergente 3D-HEVC, e explicar sobre seus métodos de codificação.

## 3.1 Estrutura de codificação

A codificação do padrão emergente 3D-HEVC é baseada no modelo *Multi-View Plus Depth* (MVD) (MERKLE, 2007a), onde cada vista de textura é associada a um mapa (ou canal) de profundidade. Com a utilização do modelo MVD, quadros com a informação de profundidade, que representam a distância da câmera até cada objeto, também devem ser gravados. Quando uma vista de profundidade estiver presente, ela necessariamente deverá estar associada a uma vista de textura, isto é, a mesma câmera será responsável por gravar tanto quadros de textura como quadros de profundidade.

Embora o padrão admita a possibilidade de codificar vídeos com um mapa de profundidade associado para cada vista, ele também pode ser utilizado para codificar vídeos sem mapas de profundidade (apenas texturas), utilizando os mesmos algoritmos de codificação, removendo apenas os algoritmos específicos para a predição de mapas de profundidade. É importante ressaltar que a codificação de mapas de profundidade não gera dependências para a codificação da vista base de textura, sendo possível fazer a decodificação da vista base de textura sem a necessidade de decodificar os mapas de profundidade. Porém, as demais vistas de textura (exceto a vista base) podem conter dependências com as vistas de profundidade (*inter-component prediction*) (TECH, 2013b).

A estrutura de codificação do 3D-HEVC é baseada em *access units*. Cada *access unit* é a estrutura composta pelo conjunto de quadros (tanto de textura como de profundidade) de um determinado instante de tempo. Os quadros de textura e os mapas de profundidade (quando presentes) são codificados de *access unit* em *access unit* (todas as vistas no mesmo instante do tempo antes de codificar outro instante de tempo), como mostrado na Figura 7 (TECH, 2013a). Neste caso, todos os quadros de um dado instante de tempo devem ser codificados antes que os dados de outro instante de tempo (tanto anterior como posterior) sejam codificados. A ordem de codificação de uma *access unit* não necessita ser exatamente na ordem temporal.

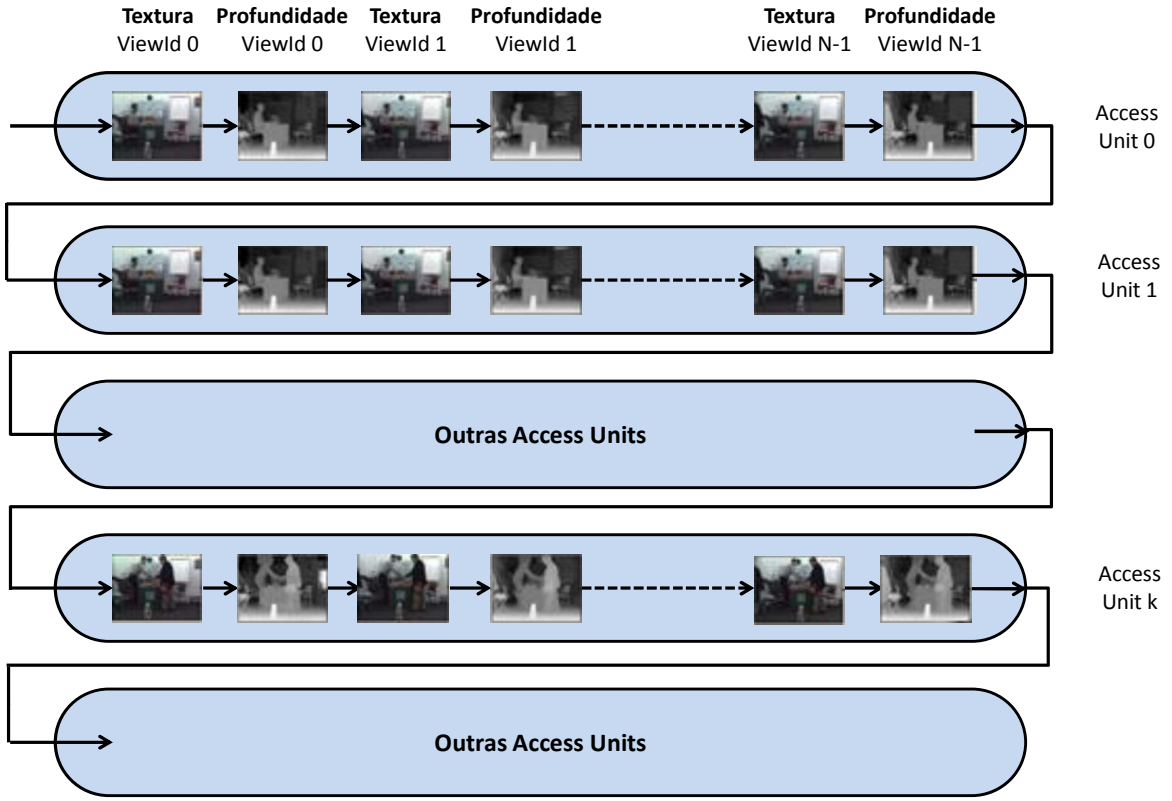



**Figura 7 – Estrutura de Access Units e ordem de codificação (TECH, 2013a).**

Os quadros de textura e de profundidade de uma câmera em particular são indicados por um identificador de câmera (*viewId* na Figura 7). Todas as imagens de textura e de profundidade de uma mesma câmera estão associadas com o mesmo valor de *viewId*. Dentro de uma *access unit*, são primeiramente codificadas as vistas com *viewId* de valor zero, depois as vistas com *viewId* de valor um e assim por diante.

A vista de textura independente (a primeira vista a ser codificada, que não tem dependência de outras vistas) é sempre codificada antes do seu mapa de profundidade associado. Para vistas dependentes, a vista de textura pode ser codificada antes ou depois do seu mapa de profundidade associado.

A estrutura básica de codificação dos vídeos 3D usando o padrão emergente 3D-HEVC é mostrada no diagrama de blocos da Figura 8. A vista independente de textura é codificada utilizando um codificador tradicional HEVC. Para fazer a codificação das vistas dependentes de textura, e mapas de profundidade, codificadores HEVC com modificações são usados. Nesses codificadores HEVC modificados, novas ferramentas para a codificação e novas técnicas de predição *inter-component* são incluídas, ou seja, técnicas que utilizam informações de textura

para a codificação de profundidade e informações de profundidade para codificação de textura. A reutilização dos dados codificados na mesma *access unit* é mostrado pelas setas vermelhas na Figura 8.

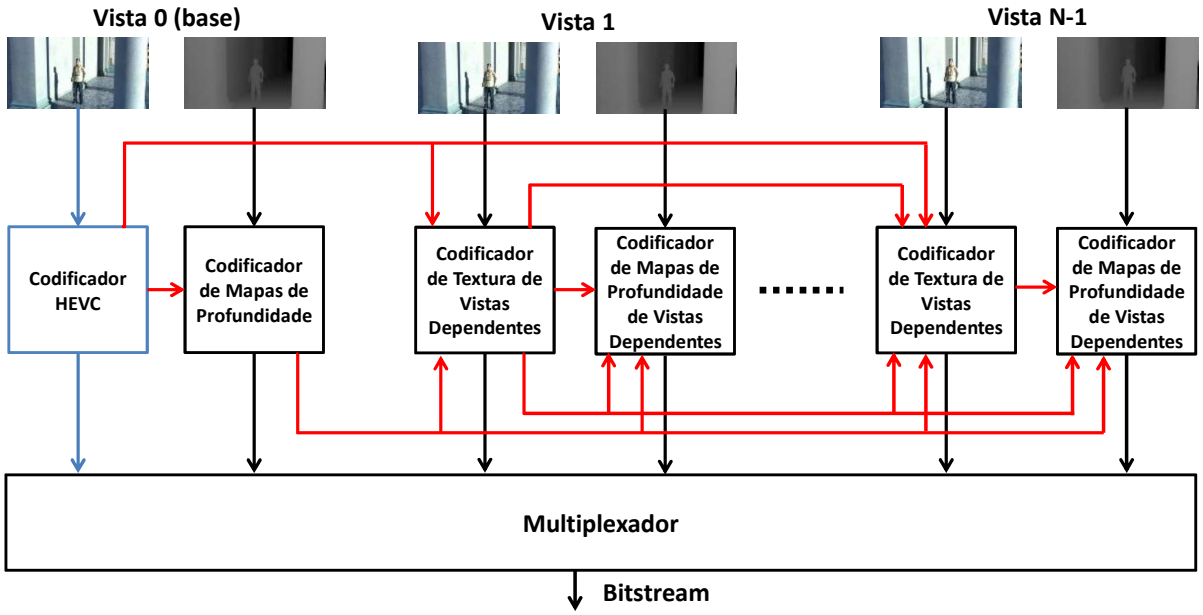



**Figura 8 – Estrutura de codificação do 3D-HEVC (TECH, 2013a).**

Um detalhe interessante que deve ser levado em conta é que os decodificadores de vídeos 2D não possuem ferramentas que decodifiquem mapas de profundidade (MULLER, 2013). Dessa forma, a vista base não pode conter dependências com mapas de profundidade.

A seção a seguir apresenta detalhes sobre as ferramentas para codificação de textura inseridas no padrão emergente 3D-HEVC. Detalhes sobre codificação de quadros de profundidade são apresentados na seção 3.3.

## 3.2 Ferramentas para a codificação de textura no 3D-HEVC

Para fazer a codificação das vistas dependentes, os mesmos conceitos presentes no padrão HEVC são utilizados, porém, são adicionadas novas ferramentas que consideram informações já utilizadas, em uma vista previamente codificada, para representar de forma mais eficiente vistas dependentes. As próximas cinco subseções apresentam algumas das novas ferramentas de codificação que estão sendo propostas pelo padrão emergente 3D-HEVC.

### 3.2.1 *Estimação de Disparidade*

Além da predição por compensação de movimento (*Motion Compensated Prediction* - MCP), que já é utilizada no HEVC, o 3D-HEVC utiliza o conceito de estimação de disparidade (*Disparity Compensated Prediction* - DCP) (GUO, 2005). A estimação de disparidade não é um conceito novo, pois já fazia parte do padrão H.264/MVC (VETRO, 2011). No entanto, o grande potencial de ganhos na predição fez com que o padrão 3D-HEVC também incorporasse esta ferramenta, que já é suportada pelo software de referência do 3D-HEVC.

O conceito é basicamente o mesmo ao MCP, porém, o MCP é aplicado entre quadros temporalmente vizinhos, enquanto o DCP é aplicado entre quadros de vistas vizinhas, mas em um mesmo instante de tempo (na mesma *access unit*). Basicamente, o DCP funciona da seguinte forma: o quadro que está sendo codificado é dividido em vários blocos e cada bloco é buscado em uma área de busca ao redor do bloco co-localizado no quadro de uma vista já codificada. Para fazer esta busca é aplicado um algoritmo com o foco em encontrar o melhor casamento de blocos possível. Quando o melhor casamento de blocos for encontrado, então um vetor de disparidade é gerado apontando para esta posição, e o resíduo entre o bloco atual e o bloco apontado pelo vetor é gerado e enviado para as próximas etapas do codificador.

Os conceitos de MCP e DCP podem ser observados na Figura 9, onde o quadro atual foi codificado com informações de até três quadros anteriormente capturados (blocos vermelho, azul e verde) usando MCP e um quadro no mesmo instante de tempo de uma vista vizinha (bloco amarelo) usando DCP.

### 3.2.2 Predição Inter-Vistas Baseada em Síntese de Vistas

Este algoritmo está sendo proposto para o padrão emergente 3D-HEVC, porém, ainda não está presente no software de referência (TECH, 2013a). O codificador e o decodificador utilizam o mesmo algoritmo para fazer a síntese de vistas, baseado na predição entre vistas. Baseado em todas as vistas já codificadas, uma nova vista virtual é sintetizada para a posição da vista que está sendo processada. Algumas regiões podem não estar presentes nessa vista sintetizada devido a oclusões. As regiões que estão presentes são marcadas em um mapa binário de disponibilidade, que controla o processo de codificação e decodificação. O codificador e o decodificador utilizam esse mapa para determinar se uma CU é

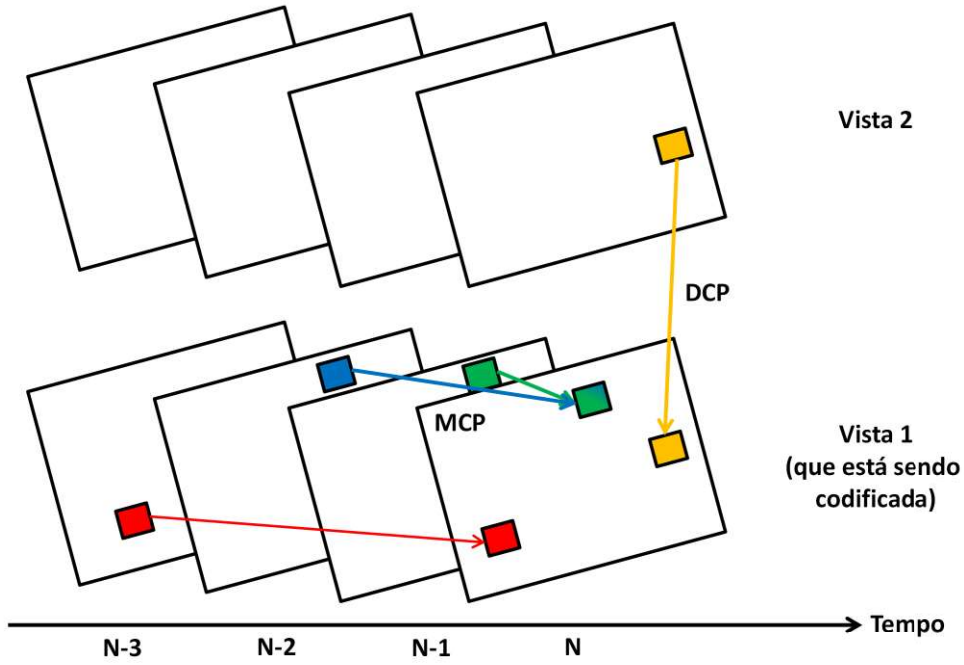



**Figura 9 – Exemplo de DCP e MCP (TECH, 2013a).**

codificada ou não. Em uma cena típica, a imagem é bastante similar entres vistas vizinhas e, dessa forma, apenas CUs em regiões com oclusões precisam ser codificadas.

Um exemplo disso é mostrado na Figura 10. A Figura 10 (a) mostra o quadro original sendo codificado. Na Figura 10 (b) aparecem as áreas oclusas entre o quadro que está sendo codificado e o quadro utilizado como referência. Na Figura 10 (c) são mostradas as CUs que necessitam codificação, onde é possível notar que esse algoritmo é bastante eficiente visto que grande parte das CUs não necessitam ser codificadas, porque as vistas apresentam poucas oclusões.

(a) (b) (c)

**Figura 10 – (a) Vista original, (b) Áreas oclusas entre duas vistas, (c) CUs a serem codificadas (TECH, 2013a).**

### 3.2.3 Loop de filtragem de pós-processamento

Esta é mais uma nova ferramenta do padrão emergente 3D-HEVC que ainda não está presente no software de referência. Este filtro é aplicado após o processo de predição de vista sintetizada para reduzir artefatos desta nova vista. Esta

ferramenta consiste em dois filtros: (1) *Depth-Gradient-based Loopback Filterer* (DGLF) *e Availability Deblocking Loopback Filter* (ADLF) (TECH, 2013b).

O DGLF reduz artefatos introduzidos pela técnica *depth-image-based rendering* (DIBR) em áreas de troca abrupta de profundidade. As vistas sintetizadas são filtradas de acordo com o gradiente da profundidade. Áreas homogêneas não são filtradas e áreas com arestas bem definidas são filtradas com fortes filtros passa baixa (OPPENHEIM, 1996) (TECH, 2013a).

O próximo filtro aplicado é o ADLF. Este busca reduzir os artefatos que são gerados como resultado da codificação por CU. Ele produz transições suaves entre áreas codificadas (enviadas no *bitstream*) e áreas sintetizadas (geradas no decodificador a partir da síntese de vistas) através de interpolações entre elas.

### 3.2.4 Compensação de Iluminação (IC)

Um modelo de compensação de iluminação é utilizado para adaptar a luminância e crominância dos blocos preditos pelo processo *inter-view* para a iluminação da vista atual. Os parâmetros do modelo linear são estimados para cada PU, usando amostras reconstruídas do bloco atual e do bloco de referência usado na predição. A compensação de iluminação também é aplicada para mapas de profundidade (TECH, 2013a). Este algoritmo já está presente no software de referência.

### 3.2.5 Ajuste do QP de textura baseado nos mapas de profundidade

Para elevar a qualidade visual das imagens de textura codificadas, uma nova ferramenta foi proposta no padrão emergente 3D-HEVC. Essa ferramenta ainda não está presente na versão atual do software de referência. A ideia básica deste algoritmo é aumentar a qualidade dos objetos no primeiro plano, e aumentar a compressão (diminuindo a qualidade) dos objetos no plano de fundo (TECH, 2013a).

Essa qualidade é ajustada quando as CUs são codificadas, modificando o parâmetro QP, que nesse caso será dependente do valor do mapa de profundidade associado a ele. Esse ajuste de QP deve ser feito tanto no codificador quanto no decodificador, sem que informações adicionais sejam enviadas no *bitstream*. Na vista base essa ferramenta é desabilitada para que se mantenha a compatibilidade com um decodificador HEVC padrão. A equação (1) apresenta como o QP de textura é modificado, onde o QP' é o valor de QP para uma CU com o valor correspondente de disparidade $d_{x,y}$.

$$QP' = QP - 2.6 + 8 \cdot \left( \frac{255 - \max\limits_{x,y \in CU} d_{x,y}}{256} \right)^2 \quad (1)$$

Esta equação, que seleciona o QP de forma adaptativa, faz com que os objetos que estão mais distantes de câmera sejam quantizados por valores maiores de QP, ou seja, reduzindo sua qualidade de vídeo e aumentando sua taxa de compressão. Por outro lado, os objetos mais próximos da câmera são quantizados por valores mais baixos de QP, ou seja, diminuindo as perdas na qualidade destes objetos. Desta forma o espectador consegue visualizar com uma maior qualidade os objetos que estão próximos à câmera, sem comprometer a taxa de compressão atingida.

## 3.3 Codificação de Profundidade

Esta seção apresenta detalhes sobre os mapas de profundidade na subseção 3.3.1, e as ferramentas para codificá-los na subseção 3.3.2.

### 3.3.1 Mapas de Profundidade

A representação do vídeo com um mapa de profundidade associado a cada quadro de textura é uma extensão de um vídeo 3D convencional, que utilizava apenas dados de textura. Esta representação é adequada para fazer síntese de vistas intermediárias com maior qualidade, já que dispensa a necessidade de inferir a profundidade a partir dos canais de textura, evitando assim os erros inerentes a este processo. Mapas de profundidade são representados utilizando informações em tons de cinza para representar a distância entre câmera e objetos, sendo que quanto maior a distância do objeto, mais escuro ele será representado. A característica de utilizar apenas tons de cinza é interessante para a codificação de vídeo já que não é necessário utilizar canais de cor adicionais. Além disso, mapas de profundidade são interessantes por permitir reduzir o número de canais de textura durante a codificação já que eles permitem uma eficiente síntese de vistas intermediárias (MERKLE, 2007b).

A Figura 11 apresenta um quadro de textura e seu respectivo mapa de profundidade associado. Através dos mapas de profundidade é possível perceber claramente os objetos presentes na imagem. Isto é uma característica importante dos mapas de profundidade, que possuem várias regiões homogêneas e arestas bem definidas (representando bordas). Os mapas de profundidade ficam restritos a

dois valores limites para a sua profundidade: $Z_{near}$ e $Z_{far}$, que representam a menor e a maior distância possíveis de serem captadas, respectivamente (MERKLE, 2007a).


**Figura 11 – Exemplo de DCP e MCP (MERKLE, 2007a).**

Os mapas de profundidade são quantizados de forma linear em amostras de 8 bits, onde o ponto mais próximo da câmera possível de ser capturado é associado ao valor de 255 (valor de $Z_{near}$), e o ponto mais distante da câmera possível de ser capturado está associado ao valor de 0 (valor de $Z_{far}$). Ao visualizar os mapas de profundidade pode-se notar que os pontos mais próximos estão representados por cores mais claras, e conforme os objetos se afastam da câmera, eles são representados por cores mais escuras. Informações detalhadas sobre o equacionamento, relacionando distâncias e os valores de profundidade, podem ser encontradas em (MULLER, 2011).

### 3.3.2 Codificação de mapas de profundidade

Para fazer a codificação dos mapas de profundidade são utilizados os conceitos de predição intra-quadro, MCP (*Motion Compensated Prediction*) e DCP (*Disparity Compensated Prediction*). Entretanto, estas ferramentas do padrão HEVC são projetadas para trabalhar sobre informações de textura, e podem não ser eficientes para codificar os mapas de profundidade. Visando o aumento da eficiência de codificação dos mapas de profundidade, algumas ferramentas foram desabilitadas ou modificadas, e novas ferramentas foram inseridas (TECH, 2013b). As próximas seis subseções apresentam as novas ferramentas, ou as modificações, propostas pelo padrão emergente 3D-HEVC para aumentar a eficiência de codificação dos mapas de profundidade.

#### 3.3.2.1 Predição por compensação de pesos $Z_{near}$-$Z_{far}$ (ZZC)

A predição por compensação de pesos $Z_{near}$-$Z_{far}$ (ZZC – $Z_{far}$ $Z_{near}$ *Compensated Weight Prediction*) é uma ferramenta nova do padrão que ainda não está descrita no software de referência. Essa ferramenta foi desenvolvida para codificação de mapa de profundidade inter-quadros (TECH, 2013b).

O conceito básico utilizado pelo ZZC é que quadros de vistas diferentes, e instantes de tempo diferentes, podem ter parâmetros $Z_{near}$ e $Z_{far}$ diferentes. Quando esses parâmetros forem diferentes entre dois quadros, então suas escalas de cinza também serão diferentes, ocasionando uma predição ruim quando um deles for utilizado como referência para o outro.

O ZZC foi projetado para resolver este problema. Antes de qualquer predição inter-quadros de profundidade, cada mapa de profundidade é redimensionado, para que ambas as escalas de cinza se refiram aos mesmos valores de profundidade.

Os valores utilizados para predição são calculados pela equação (2) (TECH, 2013a), onde $L_T$ é a disparidade compensada no *range* de profundidade $Z_{nearT}$ até $Z_{farT}$, e $L_S$ é a disparidade original no *range* de profundidade $Z_{nearS}$ e $Z_{farS}$.

$$L_T = L_S \cdot \frac{Z_{far\,S} - Z_{near\,S}}{Z_{far\,T} - Z_{near\,T}} + 255 \cdot \frac{Z_{near\,S} - Z_{near\,T}}{Z_{far\,T} - Z_{near\,T}} \quad (2)$$

Na equação (2) é feita uma compensação linear da profundidade original para uma nova profundidade compensada, obtendo um valor de profundidade relativo à profundidade de referência.

#### 3.3.2.2 Eliminação de filtros interpoladores

Os mapas de profundidade são caracterizados por arestas bem definidas e grandes regiões homogêneas (valores quase constantes). A MCP utilizada no 3D-HEVC foi modificada em relação à versão utilizada no HEVC, de forma a remover a estimação de movimento fracionária. Isto ocorre porque o filtro de interpolação de oito-*taps* do HEVC, utilizado para compensação de movimento, pode suavizar a codificação de arestas, o que produz artefatos na geração das vistas sintetizadas. Eliminando esse filtro, além de evitar este problema, também é possível diminuir a complexidade computacional do codificador e do decodificador. Esta restrição também é aplicada a DCP, que não utiliza precisão fracionária para a codificação de

mapas de profundidade. Assim, a MCP e a DCP são realizadas apenas com precisão inteira de pixel (TECH, 2013b).

#### 3.3.2.3 Desabilitando filtragem *in-loop*

Os filtros *in-loop* do HEVC foram projetados para codificar imagens de textura. Para codificar mapas de profundidade, esses filtros perdem muito a eficiência e seu custo benefício se torna baixo. Assim, para reduzir a complexidade do codificador e do decodificador, esses filtros foram desabilitados. Esses filtros são: *de-blocking filter* e *sample-adaptive loop filter* (TECH, 2013a).

#### 3.3.2.4 Predição de profundidade simplificada

A predição de profundidade simplificada (*Simplified Depth Coding* - SDC) é uma alternativa para os modos de codificação intra-quadro, que serão apresentadas na subseção 3.3.3. Para o SDC, o bloco de profundidade é predito apenas pelo modo planar ou então pelo DMM 1 de predição de profundidade, que será apresentado na subseção 3.3.3.2. O tamanho da partição do SDC é sempre *2Nx2N*, e esta informação não precisa ser transmitida no *bitstream* (TECH, 2013a).

Antes da codificação, os valores de resíduo são mapeados para valores presentes no mapa de profundidade original, usando uma *Depth Lookup Table* (DLT). Assim, os valores de resíduo podem ser codificados apenas transmitindo o índice dessa tabela, o que reduz a informação transmitida.

A vantagem de usar a DLT é reduzir o número de bits do índice residual para sequencias com um *range* de valores reduzidos de profundidade (todos os mapas de profundidade estimados onde nem todos os valores de profundidade estão presentes). O índice residual é dado pela diferença entre o índice do valor original e o índice do valor predito para profundidade.

As DLTs usam o fato de que, nos mapas de profundidade, o *range* disponível de 256 níveis de profundidade não é totalmente utilizado. Apenas um pequeno conjunto de níveis de profundidade ocorre, devido à quantização empregada. No codificador, uma DLT dinâmica é construída analisando um determinado número de quadros (um período intra). O algoritmo para a montagem da DLT é apresentado em (TECH, 2013a).

#### 3.3.2.5 Herança dos parâmetros de movimento

A ideia da herança dos parâmetros de movimento se baseia no fato de que as características de movimento dos vídeos estão associadas com o mapa de profundidade, visto que ambos são projeções da mesma cena, na mesma posição e no mesmo instante.

Para aumentar a eficiência da codificação do mapa de profundidade, esta técnica foi desenvolvida no padrão emergente 3D-HEVC. Os parâmetros de movimento do bloco de textura correspondente são adicionados como candidatos na *merge list* de PU, para o quadro de profundidade (TECH, 2013b). A *merge list* é uma lista contendo PUs vizinhas previamente codificadas da PU que está sendo codificada. O codificador seleciona que PU na *merge list* será utilizada como referência e copia as informações desta PU para a PU que está sendo codificada.

Como os quadros de textura possuem precisão de um quarto de pixel, e os mapas de profundidade possuem precisão inteira, o processo de herança de parâmetros de movimento é quantizado para o mais próximo com precisão de pixel inteiro.

#### 3.3.2.6 Predição das *quadtrees* de profundidade

A predição de *quadtree* de profundidade é inferida através da *quadtree* de textura. O particionamento da *quadtree* de profundidade é limitado pelo mesmo nível do particionamento de textura. Para uma CTU, a *quadtree* de profundidade está correlacionada com a *quadtree* de textura, então a *quadtree* de profundidade não pode apresentar mais divisões do que a *quadtree* de textura. Quando a *quadtree* de textura está dividida em *2NxN* ou *Nx2N*, o particionamento em *2NxN*, *Nx2N* e *NxN* não é feito para profundidade (MORA, 2013). Os particionamentos possíveis de textura e profundidade são apresentados na Figura 12 (TECH, 2013a).

Os algoritmos apresentados nesta seção demonstram claramente que a de codificação de mapas de profundidade é um tema novo e complexo. Por se tratar de uma técnica que pode elevar (e muito) a eficiência da codificação de vídeos 3D, estes algoritmos devem ser analisados e otimizados, a fim de obter um maior desempenho na codificação de vídeos 3D.

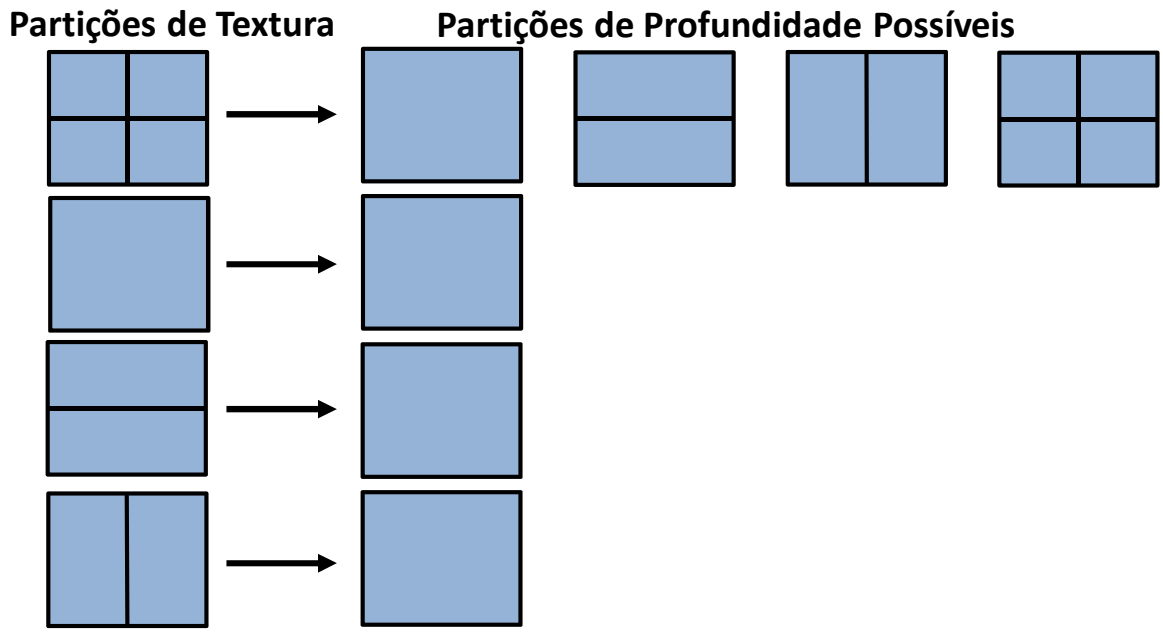



**Figura 12 – Possíveis partições de CTUs de profundidade a partir das *quadtrees* de textura adaptado de (MORA, 2013).**

### 3.3.3 Predição de mapas de profundidade

A predição intra-quadro tradicional, utilizada no padrão HEVC, explora bem as áreas homogêneas, características dos mapas de profundidade, e permite uma alta eficiência de codificação nessas áreas com valores quase constantes, porém, se ela for aplicada em uma região que contenha uma aresta, pode resultar em artefatos visíveis nas vistas intermediárias sintetizadas (MULLER, 2011).

Para contornar esse problema, o padrão emergente 3D-HEVC define quatro novos tipos de predição intra-quadro para codificação de profundidade, chamados de *Depth Modeling Modes* (DMM). Nos quatro modos, o bloco de profundidade é aproximado por um modelo que subdivide o bloco em duas regiões, onde cada região é representada por um valor constante. Esse modelo necessita de duas informações, a informação de que partição foi usada, especificando a que região cada amostra pertence, e o valor constante dessa região, que é a melhor maneira de predizer este bloco. O valor dessa região é chamado de Valor Constante de Partição (*Constant Partition Value* - CPV). No 3D-HEVC foram definidos dois tipos diferentes de partições: *Wedgelets* (ROMBERG, 2002) e Contornos (MERKLE, 2011).

Os modos de predição de profundidade foram integrados como uma alternativa para a predição intra-quadro tradicional, especificada no HEVC. Após o processo de predição, os dados residuais são enviados para as transformadas, de forma similar à predição intra-quadro tradicional.

A Figura 13 mostra uma partição *Wedgelet*. Na partição *Wedgelet*, as duas regiões são separadas por uma linha reta, que separa nas regiões $P_1$ e $P_2$. Essa separação é determinada por um ponto inicial *S* e um ponto final *E*, ambos localizados nas bordas de um bloco. No domínio discreto, a equação da reta passa pelo meio de alguns blocos, e alguns não ficariam bem definidos entre uma região ou outra. Dessa forma, é verificado se o pixel está mais presente em uma região ou outra, para selecionar a região que este bloco pertence. Com isso, cada amostra é mapeada para um valor binário, dizendo se ela pertence à região $P_1$ ou $P_2$.

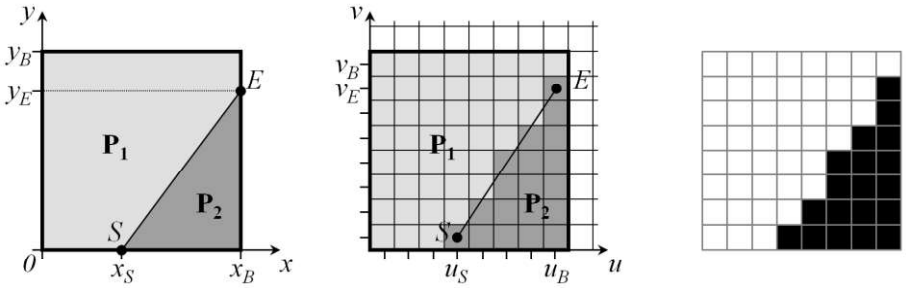



**Figura 13 – Predição de profundidade por Wedgelet (MERKLE, 2011).**

A partição por contorno é mostrada na Figura 14. Pode-se notar facilmente que ela não consegue ser descrita de forma simples por uma função matemática. Utilizando a partição por contornos, as regiões $P_1$ e $P_2$ podem ter um formato totalmente arbitrário, e até ser composta de várias regiões desconexas. Como os contornos não possuem uma linha de separação, não é necessário fazer a busca por casamento de padrão, e sim, testar pixel a pixel se está ou não dentro de uma região pré-definida.

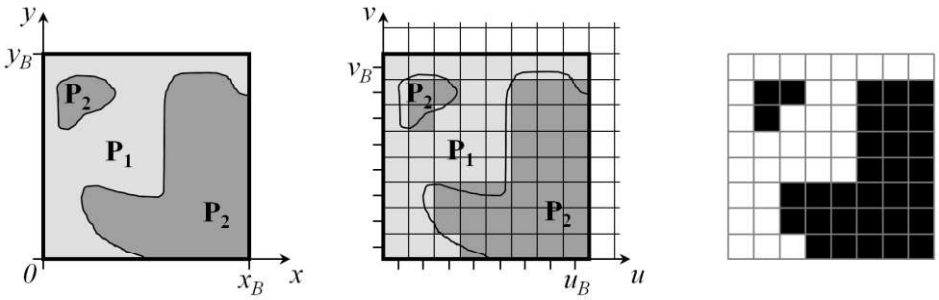



**Figura 14 – Predição de profundidade por contornos (MERKLE, 2011).**

Além da informação da partição, a segunda informação necessária para modelar o bloco de profundidade é o CPV de cada uma das regiões. Para uma dada partição, a melhor aproximação é obtida usando o valor médio do sinal original da região correspondente como CPV.

No padrão emergente 3D-HEVC foram definidos quatro modos de predição de profundidade: (1) Sinalização Explícita *Wedgelet*; (2) Predição Intra-quadro de Partições *Wedgelet*; (3) Predição *inter-component* de Partições *Wedgelet* e; (4) Predição *Inter-component* de partições de Contorno. Os modos (1), (2) e (3) utilizam *Wedgelets* para predizer o bloco, e no modo (4) é utilizado Contornos. Estes quatro modos serão apresentados nas subseções 3.3.3.2 à 3.3.3.5. Além desses modos, também foi criado outro modo com a mesma finalidade de preservar as arestas durante a codificação, chamado de codificação da cadeia limite (*Chain Coding*) que será apresentado na subseção 3.3.3.7.

### 3.3.3.1 Fluxo de codificação intra de mapas de profundidade

A Figura 15 apresenta o fluxo de codificação intra de mapas de profundidade. Inicialmente são aplicados os algoritmos de predição intra tradicional de textura, tal como previamente explicado em 2.2. Após, é aplicado o DMM 1 seguido pelo algoritmo DMM 2. Se o bloco for maior do que 4x4, também são aplicados DMM 3 e DMM 4, seguidos pelo *Chain coding*, senão, é aplicado apenas o *Chain coding*.

Ao final deste processo, é selecionado como a melhor predição intra o modo de predição que apresentar o menor custo RD.

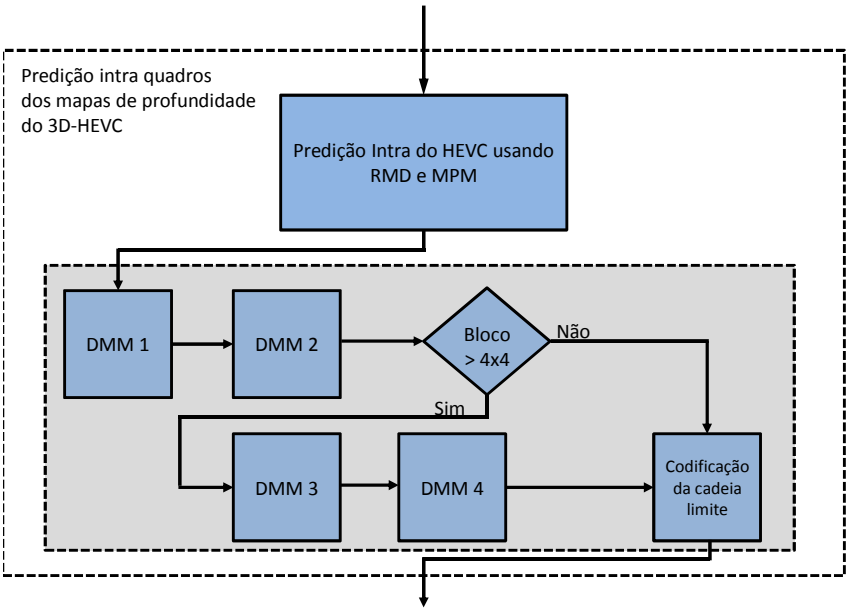



**Figura 15 – Fluxo de Codificação intra de profundidade.**

### 3.3.3.2 DMM 1 – Sinalização Explícita *Wedgelet*

Este modo procura pelo melhor casamento de partição *Wedgelet* no codificador e transmite essa informação no *bitstream*. É feita uma busca em um conjunto de possíveis partições *Wedgelets,* usando o sinal de profundidade original do bloco que está sendo codificado como referência. A busca é feita para encontrar

a partição *Wedgelet* que obtém a menor distorção entre o sinal original e a aproximação *Wedgelet* selecionada. A *Wedgelet* que obtiver o melhor resultado neste processo de predição é avaliada usando o processo do modo de decisão convencional, ou seja, fazendo o cálculo do custo do RD (TECH, 2013a).

Nesse método, um algoritmo de busca rápido é importante para diminuir a complexidade computacional do codificador. Este algoritmo de modelagem de profundidade é composto por três passos: inicialização de uma lista com padrões *Wedgelet*, busca pela menor distorção em um subconjunto desta lista e, por fim, é feito um refinamento.

No primeiro passo, ou seja, inicializar a lista com os padrões Wedgelet, é adicionado a essa lista todas as Wedgelets possíveis de serem utilizadas para cada tamanho de bloco. Após esta inicialização, um subconjunto de *Wedgelet* pré-definidos é escolhido para fazer a busca. O número de possibilidades totais de Wedgelet, assim como o número de *Wedgelets* do subconjunto utilizado na busca (antes do refinamento), são mostrados na Tabela 2 de acordo com o tamanho do bloco.

Tabela 2 – Quantidade de *Wedgelet* possíveis e avaliadas (antes do refinamento) em função do tamanho de bloco.

| **Tamanho de bloco** | **Possibilidades** | ***Wedgelets* Avaliadas** |
|---|---|---|
| 32x32 | 1503 | 368 |
| 16x16 | 1350 | 338 |
| 8x8 | 766 | 310 |
| 4x4 | 86 | 58 |

A Figura 16 apresenta todos os padrões *Wedgelets* utilizados para fazer à busca para blocos de tamanho 4x4. Para blocos de tamanho 8x8, o subconjunto cresce muito e as retas testadas são apresentadas na Figura 17. Pode-se notar que neste caso, o número de retas é elevado, e isto implica em uma grande complexidade computacional para o codificador avaliar todas estas possibilidades. Por razões de legibilidade, não serão mostradas as retas *Wedgelets* testadas para blocos de tamanho 16x16 e 32x32.

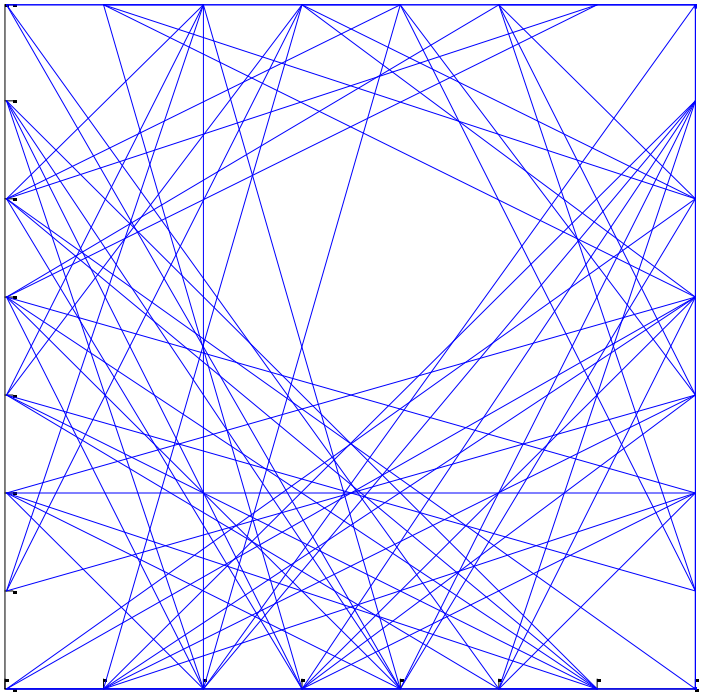


**Figura 16 – Subconjunto de *Wedgelets* avaliadas para blocos de tamanho 4x4.**

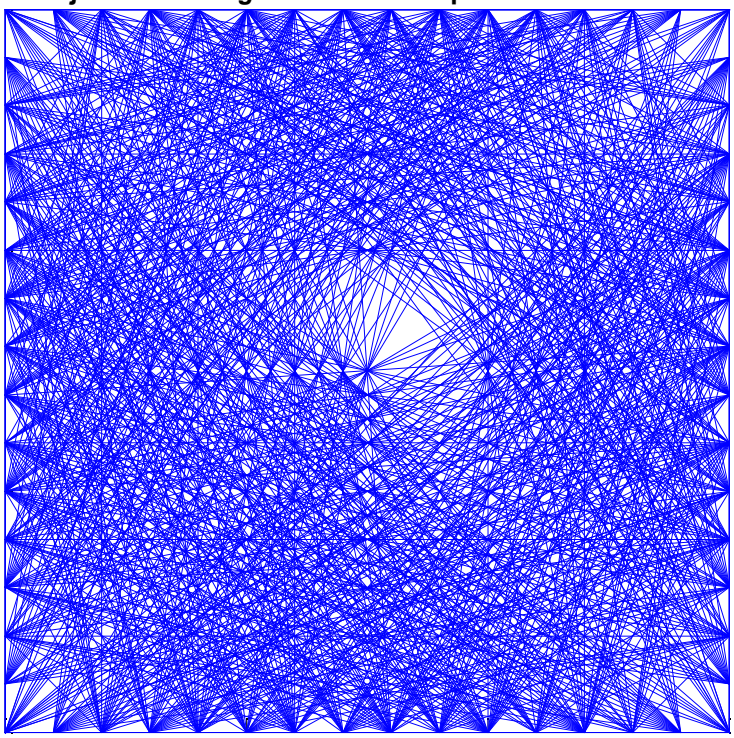


**Figura 17 – Subconjunto *Wedgelets* avaliadas para blocos de tamanho 8x8.**

Em qualquer um destes casos, as retas são testadas buscando a menor distorção. Após esta busca, uma reta *Wedgelet* é selecionada e é aplicado um refinamento. Neste refinamento são testadas até oito retas ao redor desta reta (se existir possibilidade) para verificar se alguma delas possui uma menor distorção.

#### 3.3.3.3 DMM 2 – Predição Intra-quadro de Partições *Wedgelet*

A ideia básica desse modo é predizer a partição *Wedgelet* com os dados de blocos previamente codificados dentro da mesma imagem (usando predição intra-quadro). Para uma melhor aproximação, ao final é aplicado um refinamento, variando a posição final da *Wedgelet*. Neste caso, é enviado no *bitstream* apenas o *offset* dado à posição final da linha (MULLER, 2013).

Neste modo, o processo de predição deriva a sua posição inicial e o gradiente (direção) das informações dos blocos vizinhos previamente codificados (blocos acima e a esquerda do bloco atual). Se esses vizinhos não estiverem disponíveis (por exemplo: um bloco da borda superior esquerda do quadro), o processamento deste modo é feito utilizando valores padrão. A Figura 18 mostra os dois tipos de predições feitas por esse método. A primeira, mostrada na Figura 18 (a), ocorre quando alguma referência é do tipo *Wedgelet*. O método faz o prolongamento da reta *Wedgelet* no bloco atual (só é possível se o prolongamento da *Wedgelet* intersecta o bloco atual). Se isso for possível, então os pontos iniciais $S_p$ e $E_p$ são preditos calculando o ponto de intersecção com as bordas do bloco atual.

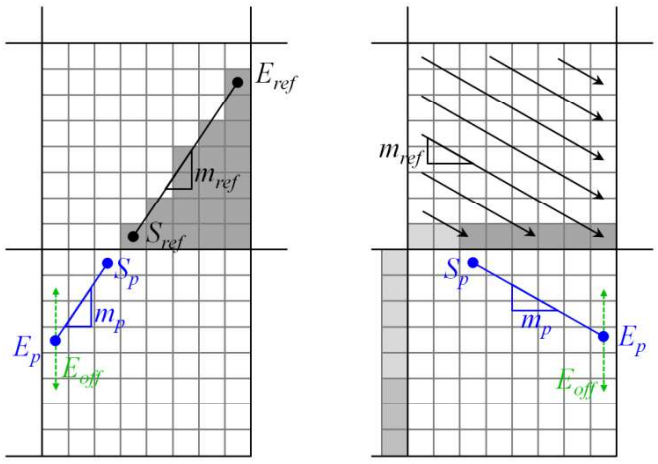



**Figura 18 – Modo 2, onde a referência é (a) *Wedgelet* e (b) intra-quadro de modo tradicional (direita) (MULLER, 2013).**

A segunda possibilidade, mostrada na Figura 18 (b), é avaliada quando as duas referências (acima e a esquerda) são do tipo intra-quadro padrão. O método inicialmente deriva a direção intra-quadro para ser usada. O ponto inicial da reta $S_p$ é derivado através de amostras adjacentes da esquerda e acima do bloco atual. É selecionada a amostra da posição com o maior gradiente. A posição final $E_p$ é

facilmente calculada através de uma equação de reta utilizando a posição inicial e a inclinação da reta.

Em ambos os métodos, a posição final da linha é refinada. Como mostrado na Figura 18, esse refinamento é feito de forma iterativa, testando as possibilidades dentro de um *range* pré-definido. O melhor resultado deste refinamento é enviado para o processo de cálculo de custo RD.

#### 3.3.3.4 DMM 3 – Predição *Inter-Component* de Partições *Wedgelet*

Este modo pode ser aplicado de duas formas. Na primeira, a predição de *Wedgelet* é feita a partir do bloco de textura utilizado como referência (o bloco co-localizado da imagem de textura associada). Este tipo de predição é chamado de *inter-component prediction* (MULLER, 2013), uma vez que ele utiliza informações da componente de textura para a predição de profundidade.

Na Figura 19 é mostrado esse tipo de predição por *Wedgelet*, em azul (parte superior). Analisando a Figura 19, pode-se notar que a direção intra-quadro de luminância está correlacionada com as direções das arestas nos quadros de profundidade. Isto motivou os especialistas a desenvolverem tanto o DMM 3, como o DMM 4, utilizando a técnica de *inter-component prediction*.

A primeira forma de usar a predição do Modo 3 é utilizando a direção intra-quadro do bloco co-localizado, visando restringir o conjunto das possíveis partições *Wedgelets*, e fazer a busca apenas pelas partições *Wedgelets* mais prováveis. Essa busca é realizada em duas partes: na primeira parte, para cada direção intra-quadro, uma lista restrita de *Wedgelets* é inicializada com os padrões *Wedgelets* que apresentam direções similares a intra-quadro. Na segunda parte, apenas as *Wedgelets* presentes nessa lista são testadas. Após verificar qual Wedgelet obteve o melhor resultado, é necessário transmitir apenas o índice da lista *Wedgelet* para o decodificador compreender as regiões de separação do bloco. Mais detalhes sobre a geração da lista para as direções intra-quadro são apresentados em (TECH, 2013a).

A segunda forma do Modo 3 é utilizada quando o bloco co-localizado não é codificado com intra-quadro. Neste caso, as partições *Wedgelet* são restritas ao subconjunto de partições utilizadas no Modo 1, porém, sem aplicar o refinamento neste modo.

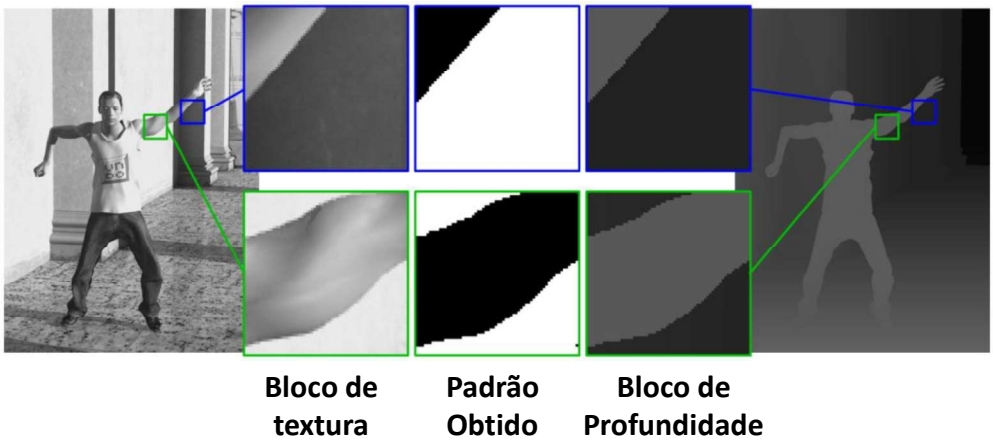



**Figura 19 – Predição por *Wedgelet* (em azul) e por cortorno (em verde) através de *inter-component prediction* (TECH, 2013a).**

#### 3.3.3.5 DMM 4 - Predição *Inter-Component* de Partições de Contorno

A ideia principal deste método é predizer a partição de contornos a partir de um bloco de textura de referência (o bloco co-localizado). Este processo, tal como o Modo 3 é feito utilizando *inter-component prediction*, e está representado na Figura 19 em verde. Pode-se notar que a região do braço tanto na parte de textura (luminância apenas) como na parte de profundidade, apresentam características muito similares. Com isso, utilizar essa redundância entre as componentes é interessante para obter melhores resultados, motivando os especialistas a desenvolverem o algoritmo do DMM 4.

Na predição do DMM 4, o canal de textura é utilizado como referência, utilizando o sinal de luminância reconstruído do bloco de textura co-localizado ao bloco de profundidade que está sendo codificado. A estratégia do DMM 4 é aplicar um critério de *threshold* na predição da partição. Neste caso, o *threshold* utilizado é o valor médio do sinal de luminância do bloco co-localizado de textura (esse valor médio é representado pela variável $u$). As regiões são particionadas como segue: as amostras do bloco de textura maiores do que a média, são atribuídas à região $P_1$, enquanto as amostras menores ou iguais são atribuídas à região $P_2$, tal como mostrado em (3), onde *p* representa o valor da amostra de textura.

$$\begin{cases} P_1 & se \quad p < u \\ P_2 & \quad senão \end{cases} \tag{3}$$

Finalmente, os valores médios de cada região das amostras de profundidade são utilizados como predição para cada uma das regiões e então, as amostras originais de profundidade são subtraídas das preditas e o resíduo é gerado.

#### 3.3.3.6 Codificação de Valor Constante de Partição (CPV)

O método de codificação de CPV é o mesmo nos quatro modos de predição de profundidade. Como mostrado na Figura 20, este método utiliza três informações para definir o CPV: o CPV original, o CPV predito e delta CPV (MERKLE, 2011).

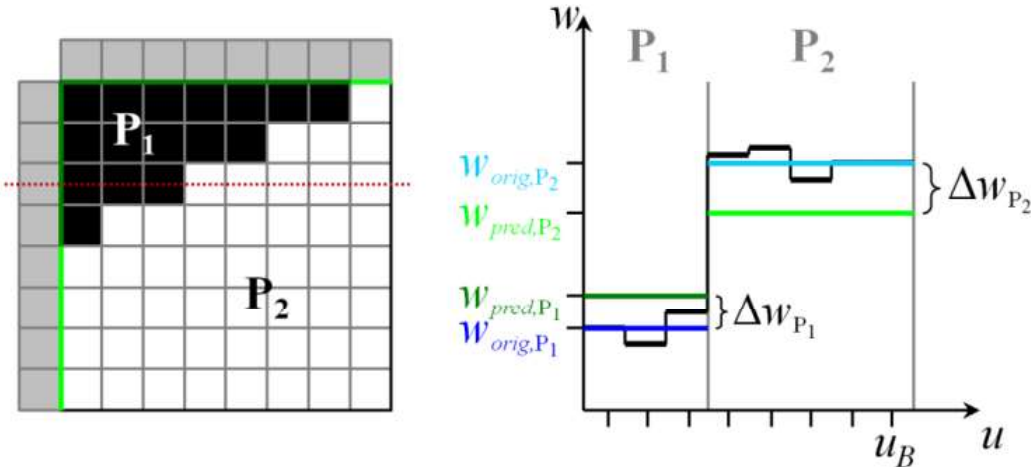



**Figura 20 – Codificação de CPV (MERKLE, 2011).**

Os valores dos CPVs originais são calculados como o valor médio do sinal em cada uma das regiões ($P_1$ ou $P_2$). Estes valores não estão presentes no decodificador, pois eles necessitam do sinal original, logo, eles são preditos. O erro de predição pode ser corrigido através da transmissão do delta CPV. Os CPVs preditos são derivados das amostras dos blocos vizinhos adjacentes (à esquerda e acima). Isto é mostrado na Figura 20 à esquerda, onde o verde escuro mostra as amostras adjacentes à partição $P_1$, e o verde claro mostra as amostras adjacentes à partição $P_2$.

Além disso, é possível transmitir o delta CPV quantizado para reduzir a largura de banda. Na Figura 20 é possível observar os valores escolhidos para os CPVs originais e os CPVs preditos, de acordo com a intensidade das amostras de bloco.

#### 3.3.3.7 Codificação da região da cadeia limite (*Chain Coding*)

O método de codificação do *Chain Coding* limite divide o bloco em duas regiões, sinalizando os limites de cada região com codificação de cadeias. Este método consiste em quatro etapas que são: (1) encontrar arestas internas; (2) codificar as arestas usando codificação de cadeias; (3) converter a codificação de cadeia no *bitstream*; (4) calcular as predições (TECH, 2013a).

No primeiro passo, as arestas internas são encontradas. Para isso, primeiramente são calculadas as diferenças entre os pixels adjacentes na vertical e horizontal. Se essa diferença for maior que um determinado *threshold*, então essa aresta é marcada como candidata. Este processo é feito para selecionar como possíveis candidatos apenas arestas. No próximo passo, as arestas candidatas cuja diferença for menor do que as diferenças das arestas vizinhas são eliminadas do processo. Se alguma aresta estiver desconexa, então é feita a ligação. Por fim, é feita uma checagem se o bloco contem exatamente duas regiões. Caso o bloco não contenha exatamente duas regiões, este processo é descartado.

No segundo passo, as arestas são codificadas usando codificação de cadeias. Inicialmente, o algoritmo começa de uma borda do bloco. Então, a próxima aresta é escolhida como uma aresta vizinha até que chegue a uma aresta final (em um limite do bloco). Para construir a codificação de cadeia, são definidos sete ângulos (0, 45, -45, 90, -90, 135 e -135 graus). A Figura 21 mostra esses ângulos quando o pixel anterior é posicionado a esquerda.

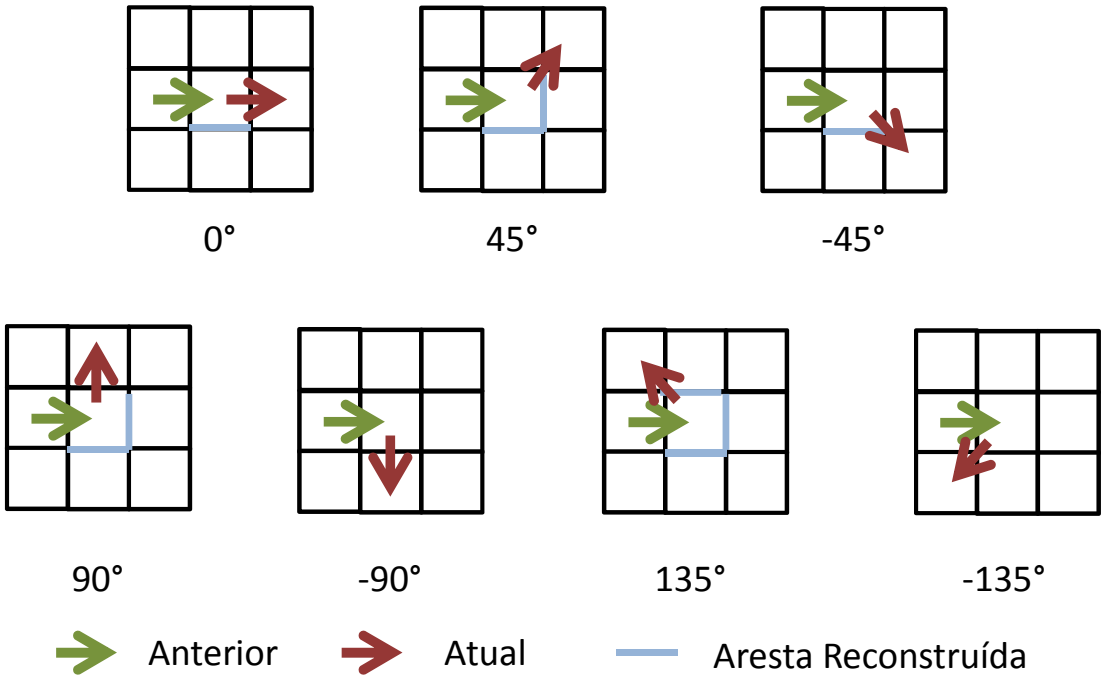



**Figura 21 – Construção das arestas usando *Chain Coding* adaptado de (TECH, 2013a).**

No terceiro passo, a cadeia de codificação é convertida em uma sintaxe, expressando o ponto inicial da cadeia e as retas que formam essa cadeia. Por fim, o valor médio de cada região é escolhido como valor predito para cada uma das

regiões. Neste passo, pode-se aplicar codificação de CPV tal como mostrado na subseção 3.3.3.6.

Neste trabalho, estamos considerando a versão 7.0 do HTM, entretanto, considerando a versão mais nova foram removidos os modos DMM 2, DMM 3 e o *Chain Coding* por apresentarem ganhos baixos em relação a sua complexidade computacional em termos de tempo de codificação.

Neste capítulo foi apresentada a estrutura de codificação do 3D-HEVC e os principais novos algoritmos utilizados para a codificação de textura e mapas de profundidades. Através deste capítulo, pode-se perceber que existe uma grande quantidade de novos algoritmos inseridos neste padrão, resultando em um aumento da complexidade comparado com antigos codificadores 3D. No próximo capítulo são apresentados os principais desafios relativos a codificação do padrão 3D-HEVC e também, os algoritmos desenvolvidos pelos trabalhos relacionados na literatura.

# 4 ESTADO DA ARTE E DESAFIOS DE PESQUISA

Neste capítulo são apresentados os principais trabalhos relacionados presentes na literatura, propondo soluções para a codificação de vídeos 3D, onde são apresentadas as principais limitações relativas a estes trabalhos. Além disso, este capítulo apresenta os principais desafios de pesquisa relativos à codificação de vídeo 3D com mapas de profundidade associados.

## 4.1 Estado da Arte

Existem vários trabalhos relacionados na literatura propondo esquemas de redução de complexidade para a codificação de vídeos 3D. Os trabalhos (KIM, 2007), (LI, 2008), (ZHU, 2010), (SHEN, 2009), (SHEN, 2010), (KUO, 2010), (YANG, 2010), (HE, 2009), (ZHANG, 2012), (ZATT, 2010) e (YEH, 2014) focam no atual padrão estado da arte de codificação de vídeos 3D, o H.264/MVC. Em (KIM, 2007), (LI, 2008), (ZHU, 2010) e (YEH, 2014), técnicas para acelerar a estimação de disparidade e de movimento são propostas. Numerosos trabalhos propõem redução de complexidade através de técnicas de *early termination*, tal como (SHEN, 2009), (SHEN, 2010), (KUO, 2010), (YANG, 2010), (HE, 2009), (ZHANG, 2012) e (ZATT, 2010), onde o ganho de redução de complexidade é obtido por tomar uma decisão baseada em alguma heurística para reduzir o número de blocos ou modos avaliados. Os métodos citados em todos estes artigos podem ser aplicados para reduzir a complexidade do padrão 3D-HEVC, porém, as novas ferramentas inseridas no padrão 3D-HEVC devem ser analisadas para a proposição de um esquema para a redução de complexidade.

### 4.1.1 Redução da complexidade no 3D-HEVC

Foram encontrados quatro trabalhos na literatura atual com o foco em reduzir a complexidade especificamente do padrão 3D-HEVC. O trabalho (MORA, 2013) propõe uma técnica para limitar a *quadtree* do 3D-HEVC, tanto de textura como de profundidade. A técnica proposta por (MORA, 2013) foi apresentada em 3.3.2.6. Na versão do HTM utilizada neste trabalho, este algoritmo está presente por padrão. Desta forma, em todas nossas simulações este algoritmo estará presente e integrado com as técnicas propostas, não servindo de base para comparação.

Como mencionado em 3.3.2.6, (MORA, 2013) apresenta uma simplificação de alto nível nas *quadtrees* de profundidade. Esta técnica não apresenta nenhum

esforço para reduzir a complexidade especificamente dos novos modos de predição para mapas de profundidade, que foram inseridos neste padrão, mas simplesmente utiliza uma solução para a redução de complexidade em um nível mais global. Desta forma, existe um vasto espaço para explorar novas soluções que podem ser utilizadas em conjunto com essa.

O trabalho de (SONG, 2013) visa reduzir o número de *Wedgelets* avaliadas no decodificador pelo DMM 3 aplicando um filtro nas bordas do bloco de textura correlacionado ao bloco de profundidade que está sendo decodificado. Dessa forma, (SONG, 2013) consegue detectar as Wedgelets mais prováveis de serem utilizadas durante o processo de decodificação. O algoritmo apresentado em (SONG, 2013) calcula o gradiente de todas as bordas do bloco de textura e verifica os pontos com maior probabilidade de iniciar ou terminar uma *Wedgelet* na textura, reduzindo o número de Wedgelets avaliadas na decodificação para apenas seis. Com isso, (SONG, 2013) consegue reduzir a complexidade da decodificação em média em 3,1%. Como (SONG, 2013) foca no decodificador, não será possível realizar comparações com o nosso trabalho porque todos os algoritmos presentes nesta dissertação são focados no codificador.

Em (GU, 2013a) foi publicado um trabalho que posteriormente foi ampliado e submetido em (GU, 2013b) para ser incorporado ao HTM, quando foi aceita sua incorporação. A partir deste ponto, será citada apenas a referência a (GU, 2013b).

O trabalho apresentado em (GU, 2013b) propõe um algoritmo para reduzir a complexidade da codificação intra de mapas de profundidade do 3D-HEVC através da redução da complexidade dos modos DMM. O algoritmo proposto descarta a codificação DMM quando o primeiro modo intra a ser avaliado pelo custo RD é o modo planar. Esta técnica funciona bem porque ao aplicar o algoritmo RMD para selecionar os modos que serão avaliados na predição intra tradicional, se o modo planar for o primeiro a ser escolhido, a tendência é que o bloco tenha características homogêneas. Além disso, o algoritmo proposto por (GU, 2013b) analisa a variância do bloco de profundidade que está sendo codificado. Caso a variância do bloco seja elevada, então os modos DMM são utilizados para a codificação, caso contrário os modos DMM são cortados. É oportuno informar que este algoritmo foi inserido na versão do HTM posterior a usada neste trabalho, o HTM-8.0.

O algoritmo de (GU, 2013b) consegue atingir uma redução de complexidade (em termos de tempo de codificação) de apenas 2,5% em relação ao HTM-7.0, com uma diminuição de BD-rate de 0,01%, o que significa que para uma mesma qualidade de vídeo o *bit rate* foi reduzido em 0,01%. Além disso, considerando a redução de complexidade obtida em (GU, 2013b), é possível atingir uma maior redução de complexidade através de uma solução mais sofisticada.

O trabalho de (ZHANG, 2014) propõe utilizar informações de das quadtrees de textura para determinar quando as quadtrees estão em regiões simples ou em regiões complexas. Ao determinar que uma quadtree está em uma região simples, (ZHANG, 2014) remove as avaliações dos modos DMM e faz apenas a predição intra tradicional, enquanto se a região for classificada como região complexa, então ele executa tanto a predição intra quanto os novos modos DMM.

Este algoritmo foi avaliado no HTM-5.1 para a configuração All-Intra (AI). Em média, ele consegue obter uma redução de complexidade de 40,3% com perdas de BD-rate médias de 0,9%.

## 4.2 Principais Desafios

Uma série de experimentos foi realizada para determinar a complexidade inserida no padrão 3D-HEVC para fazer a codificação de mapas de profundidade. Para isso, todas as sequências de teste presentes nas Condições Comuns de Teste (CCT), que serão descritas posteriormente na seção 6.1, foram utilizadas nas simulações com o HTM-7.0, e os tempos para codificar cada quadro de textura e de profundidade foram salvos. Com isso, foi possível perceber que a custo computacional da codificação dos mapas de profundidade representa 24,3% do custo do codificador completo, enquanto a codificação de textura representa os outros 75,7%, como mostrado na Figura 22.

É oportuno ressaltar que esta complexidade foi adicionada ao codificador completo, visto que, até o momento, os codificadores de vídeo 3D trabalhavam em um sistema de codificação sem mapas de profundidade associados aos quadros de textura. Adicionar esta complexidade ao codificador, que já é bastante complexo, torna o sistema ainda mais custoso computacionalmente. De um lado, a parcela de tempo de textura é maior, representando 75,7% do tempo, entretanto, já existem diversas soluções na literatura para reduzir este problema. Por outro lado, a codificação de profundidade, mesmo representando um percentual menor de tempo

computacional, foi pouco explorada na literatura e, dessa forma, este ainda é um campo aberto para gerar novas contribuições para a redução da complexidade.

Além do exposto acima, também foi analisada, detalhadamente, a divisão dos tempos dentro dos quadros de profundidade, e os resultados desta análise são apresentados na Figura 22 (à direita). O DMM 1 é o método intra-quadro mais complexo para mapas de profundidade, representando 15,75% da complexidade do processo de codificação dos mapas de profundidade, sendo que a predição intra tradicional é o segundo mais complexo, contribuindo com 9,73% da complexidade total. A soma de todos os modos DMM, e do *Chain Coding* (métodos específicos para predição intra-quadro de profundidade), representa 25,66% da complexidade da codificação dos mapas de profundidade. A parcela "Outros", que representa 64,61% da complexidade da codificação dos mapas de profundidade, considera o esforço necessário para a codificação dos modos *skip*, predição inter-quadros, predição inter-vistas, entropia, entre outros. Como estas etapas não são o foco deste trabalho, elas não foram analisadas em maiores detalhes.

Mesmo representando um acréscimo na complexidade de um codificador de vídeo, os novos métodos de codificação intra-quadro específicos para profundidade foram inseridos no 3D-HEVC com o objetivo de melhorar a qualidade da codificação, principalmente em regiões que apresentem arestas, possibilitando uma predição eficiente de bordas e reduzindo a geração de artefatos nas vistas sintetizadas. Por outro lado, para as regiões homogêneas, a predição intra tradicional consegue obter ótimos resultados.

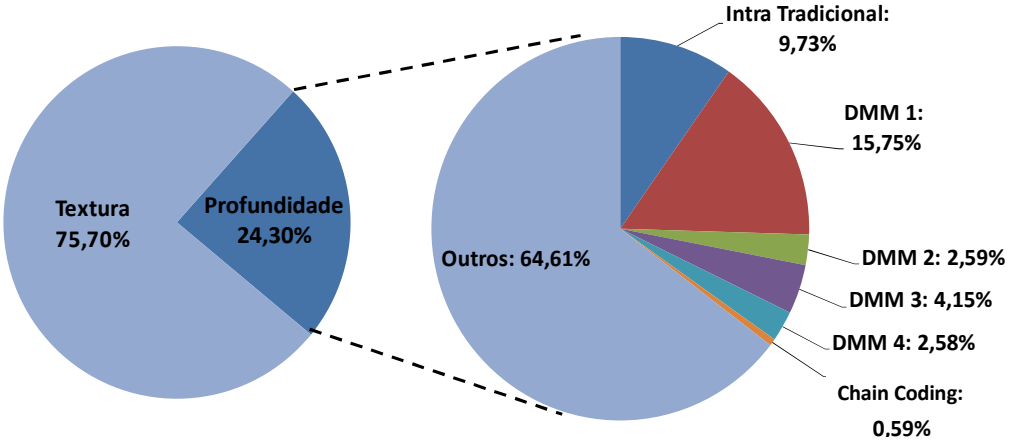



**Figura 22 – Distribuição dos tempos de execução no padrão 3D-HEVC.**

Através de outras simulações, cujos resultados são apresentados na Figura 23, foi observado que os modos DMM e *Chain Coding* são escolhidos como melhor método para codificar apenas 1,22% dos blocos (geralmente possuem arestas), enquanto a predição intra tradicional é escolhida nos outro 98,78% das vezes (áreas homogêneas). Mesmo sendo escolhidos apenas em 1,22% dos casos, os DMMs são muito importantes, pois sua remoção acarreta em uma perda superior a 2% de taxa de compressão para uma mesma qualidade de vídeo. Considerando isso, e a distribuição dos tempos na codificação de profundidade, apresentado na Figura 22, é possível notar que mesmo os DMMs e o *Chain Coding* sendo avaliados para todos os blocos, elas apresentam uma baixa probabilidade de serem escolhidas como a melhor predição possível. Em outras palavras, os modos DMM possuem uma grande importância para a eficiência da codificação, visto que eles têm a características de conseguir preservar arestas, entretanto, eles possuem uma baixa probabilidade de serem escolhidos e demandam uma considerável parcela de tempo da codificação do 3D-HEVC. Por esse motivo, algoritmos para reduzir a complexidade geral da codificação intra de mapas de profundidade, através de uma rápida decisão (não computar DMM e *Chain Coding* quando desnecessário) são muito desejados. Essas técnicas para redução de complexidade são ainda mais importantes quando lidando com sistemas de tempo real para codificação, aplicados principalmente a sistemas embarcados, visto que a complexidade inserida para a codificação de profundidade contribui para o aumento do consumo de energia.

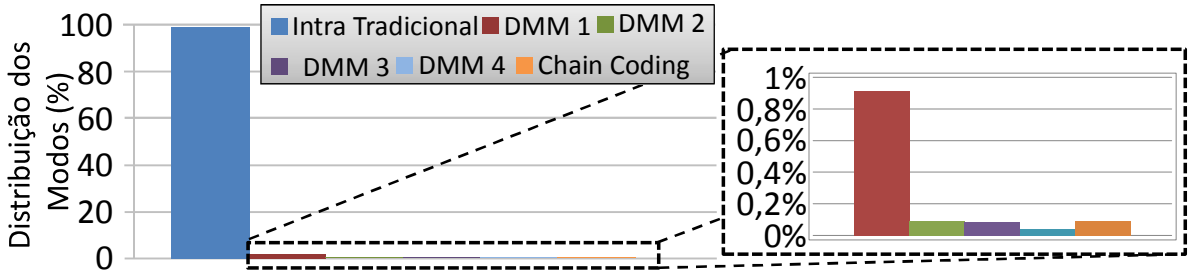



**Figura 23 – Distribuição da seleção dos modos de predição intra-quadro na codificação de profundidade.**

Considerando este cenário em que os trabalhos relacionados atacam bastante à codificação de textura e poucos trabalhos são focados na redução de complexidade para profundidade, novas técnicas para a redução de complexidade de mapas de profundidade são altamente desejáveis, entretanto, é importante que essas novas

técnicas, além de reduzir a complexidade da codificação, apresentem impactos mínimos (ou nulos) na qualidade do vídeo 3D codificado (textura e vistas sintetizadas) e na taxa (*bit rate*).

# 5 ESQUEMA PARA REDUÇÃO DE COMPLEXIDADE EM MAPAS DE PROFUNDIDADE

Este capítulo apresenta o *Depth Modeling Modes Fast Prediction* (DMMFast) que visa reduzir a complexidade computacional da codificação de mapas de profundidade, sendo composto por dois algoritmos: o *Simplified Edge Detector* (SED) e o *Gradient-based Mode One Filter* (GMOF). O SED visa reduzir a complexidade dos mapas de profundidade fazendo uma decisão se os DMMs devem ou não ser codificados, enquanto o GMOF avalia as bordas do bloco de profundidade para encontrar as posições mais promissoras de serem avaliadas pelo DMM 1. Para a definição desta estratégia foi realizado um amplo estudo sobre a codificação de mapas de profundidade no padrão emergente 3D-HEVC. A análise estatística que foi utilizada para tomar as decisões sobre o esquema de redução de complexidade proposto também é apresentada detalhadamente.

## 5.1 Análise Qualitativa

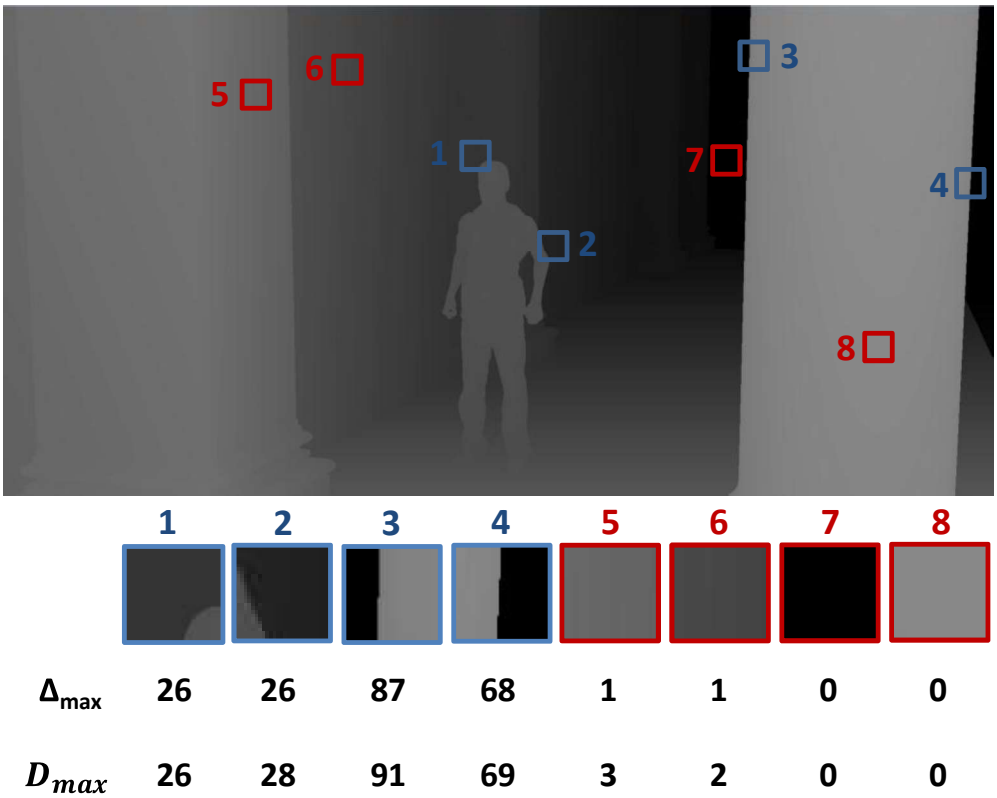
Nesta seção serão apresentadas as análises qualitativas realizadas, visando determinar as principais características de um bloco contendo uma aresta e do modo DMM 1. A análise apresentada nesta seção é crucial para o entendimento dos fatores que motivaram a criação do *Depth Modeling Modes Fast Prediction* (DMMFast) para a redução de complexidade da predição intra-quadro dos mapas de profundidade.

### 5.1.1 Classificação de blocos de profundidade

Analisando a predição intra-quadro de mapas de profundidade do 3D-HEVC, é possível notar que vários modos são avaliados, no entanto, muitos deles possuem uma baixa probabilidade de serem escolhidos como o melhor caso. Considerando as características de mapas de profundidades mencionadas em 3.3 que os mapas de profundidades possuem regiões homogêneas e arestas bem definidas, além disso, considerando que os novos modos de codificação de profundidade foram propostos com o objetivo de elevar a qualidade na codificação de arestas, intuitivamente, as operações poderiam ser simplificadas caso fosse possível classificar o bloco como aresta ou região constante. Neste caso, os DMMs seriam aplicados apenas nos casos em que o bloco é classificado como uma aresta (observando o fato de que os modos DMM foram criados para melhor codificar blocos contendo arestas).

Para ilustrar esta classificação de bloco, a Figura 24 apresenta um mapa de profundidade da sequência *Undo_Dancer* (RUSANOVSKYY, 2013), onde quatro blocos contendo arestas foram destacados em caixas azuis e blocos contendo regiões homogêneas foram destacadas em caixas vermelhas. Um *zoom* destes blocos destacados é mostrado na parte inferior da Figura 24.

Por inspeção visual dos blocos que representam regiões homogêneas (caixas vermelhas), pode-se concluir que não existem diferenças significativas entre amostras vizinhas do bloco. Considerando o caso de estudo apresentado na Figura 24, a máxima diferença entre pixels vizinhos ($\Delta_{max}$) é de 1 (valor absoluto que pode variar entre 0 e 255) para os blocos 5 à 8. Para os blocos que representam arestas (caixas azuis), existe sempre ao menos uma diferença significativa entre amostras vizinhas ($\Delta_{max}$ variando de 26 até 87 para os blocos 1 à 4).




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $\Delta_{max}$ | 26 | 26 | 87 | 68 | 1 | 1 | 0 | 0 |
| $D_{max}$ | 26 | 28 | 91 | 69 | 3 | 2 | 0 | 0 |

**Figura 24 – Mapa de profundidade da sequência *Undo_Dancer* com arestas e áreas homogêneas destacadas.**

Uma métrica que calcula todas as diferenças entre amostras vizinhas em um bloco possui um custo computacional relativamente elevado. Buscando uma métrica simplificada para este processo, foi definida a métrica $D_{max}$ para medir a máxima diferença absoluta entre os cantos dos blocos tal como expresso na equação (4). A

ideia nesta métrica é que se um bloco for praticamente constante, a diferença entre os cantos será praticamente zero, enquanto que se o bloco possuir uma única descontinuidade, então a diferença será aproximadamente o valor desta descontinuidade (o que em geral será um valor elevado).

Em casos reais, as amostras não são perfeitamente constantes, mas possuem uma diferença muito pequena para regiões praticamente constantes. Assim, a diferença entre os cantos de um bloco é uma métrica simples e bastante confiável para fazer a classificação deste bloco em região constante ou aresta. A equação (5) computa o operador *max* (o maior valor entre os operandos) entre $D_S$ (superior), $D_I$ (inferior), $D_E$ (esquerda), $D_D$ (direita), $D_{DP}$ (diagonal principal) e $D_{DS}$ *(diagonal secundária)*, que representam a diferença absoluta entre os cantos superiores (equação (5)), os cantos inferiores (equação (6)), os cantos da esquerda (equação (7)), os cantos da direita (equação (8)), os cantos da diagonal principal (equação (9)) e os cantos da diagonal secundário (equação (10)), respectivamente.

$$\mathrm{D_{max}} = \max(\mathrm{D_S}, \mathrm{D_I}, \mathrm{D_E}, \mathrm{D_D}, \mathrm{D_{DP}}, \mathrm{D_{DS}}) \quad (4)$$

$$\mathrm{D_S} = |\mathrm{P}(1,1) - \mathrm{P}(1, \mathrm{size})| \quad (5)$$

$$\mathrm{D_I} = |\mathrm{P}(\mathrm{size}, 1) - \mathrm{P}(\mathrm{size}, \mathrm{size})| \quad (6)$$

$$\mathrm{D_E} = |\mathrm{P}(1,1) - \mathrm{P}(\mathrm{size}, 1)| \quad (7)$$

$$\mathrm{D_D} = |\mathrm{P}(1, \mathrm{size}) - \mathrm{P}(\mathrm{size}, \mathrm{size})| \quad (8)$$

$$\mathrm{D_{DP}} = |\mathrm{P}(1,1) - \mathrm{P}(1, \mathrm{size})| \quad (9)$$

$$\mathrm{D_{DS}} = |\mathrm{P}(1, \mathrm{size}) - \mathrm{P}(\mathrm{size}, 1)| \quad (10)$$

Onde *size* indica o tamanho do bloco (tanto na horizontal como na vertical).

Nas regiões constantes (blocos em vermelho na Figura 24), o $D_{max}$ é desprezível, dado que as bordas do bloco apresentam praticamente valores homogêneos. Neste contexto, é possível classificar o bloco de acordo com o valor de $D_{max}$. Se o $D_{max}$ é um valor significativo (de acordo com um limiar), então o bloco pode ser classificado como aresta, caso contrário, o bloco pode ser classificado como região constante. No caso de estudo apresentado na Figura 24, todos os blocos classificados como arestas apresentaram $D_{max}$ superior a 26, enquanto blocos constantes apresentaram $D_{max}$ inferiores a 3.

### 5.1.2 Análise das bordas dos blocos de profundidade

A Figura 25 apresenta um exemplo de bloco 8x8 de um mapa de profundidade e também, a melhor *Wedgelet* selecionada a partir do uso do algoritmo DMM 1. Além disso, o gradiente de todas as bordas também são apresentados na Figura 25. Os gradientes da linha superior, da coluna esquerda, da linha inferior e da coluna direita são obtidos aplicando as equações (11), (12), (13) e (14), respectivamente. O parâmetro x nestas equações varia de 1 até *size* - 1.

$$\mathrm{Grad}_{\mathrm{Upper}}(x) = |P(1,x) - P(1,x+1)| \quad (11)$$

$$\mathrm{Grad}_{\mathrm{Left}}(x) = |P(x,1) - P(x+1,1)| \quad (12)$$

$$\mathrm{Grad}_{\mathrm{Lower}}(x) = |P(\mathrm{size},x) - P(\mathrm{size},x+1)| \quad (13)$$

$$\mathrm{Grad}_{\mathrm{Right}}(x) = |P(x,\mathrm{size}) - P(x+1,\mathrm{size})| \quad (14)$$

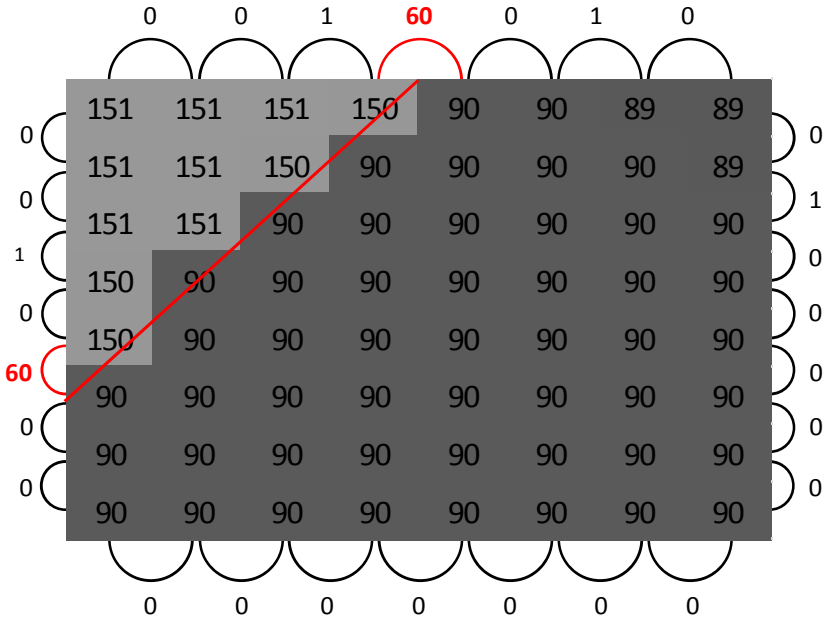



**Figura 25 – Analise dos Gradientes de um bloco de mapas de profundidade.**

Pode-se notar na Figura 25 que a melhor *Wedgelet* obtida foi escolhida iniciando (ou terminando) no maior gradiente obtido na borda superior. Como o algoritmo DMM 1 objetiva encontrar a *Wedgelet* que melhor separa o bloco em duas regiões homogêneas, o maior gradiente entre as bordas tende a ser um bom candidato para separar os blocos em duas regiões homogêneas, pois ele apresenta a melhor decisão local considerando apenas as bordas do bloco.

Existem muitas *Wedgelets* que são avaliadas no processo do DMM 1 que possuem uma baixa probabilidade de serem escolhidas (observadas na Figura 16 e

na Figura 17). Estas representam um aumento desnecessário na complexidade da codificação de vídeo. Com estas observações, é possível concluir que limitando a busca pelas *N* posições com os maiores gradientes, seria possível reduzir a complexidade do DMM 1, e ainda assim, encontrando uma partição *Wedgelet* que potencialmente pode ser ótima, ou muito próxima.

Ao utilizar o gradiente das amostras da borda para definir as posições mais promissoras para esta avaliação, é possível restringir o número de *Wedgelets* avaliadas, reduzindo a complexidade computacional do DMM 1. Além disso, ao manter a avaliação das *Wedgelets* mais promissoras, é possível também manter a qualidade da predição. Utilizando esta ideia, proposta para reduzir a complexidade do DMM 1, em conjunto com a proposta da avaliação dos cantos dos blocos, foi proposto o *Depth Modeling Modes Fast Prediction* (DMMFast), estratégia que visa reduzir a complexidade da codificação intra-quadro de mapas de profundidade, que é uma das principais contribuições deste trabalho, e será apresentado na próxima seção.

## 5.2 Depth Modeling Modes Fast Prediction (DMMFast)

O DMMFast visa reduzir a complexidade dos modos DMMs, que representam, tal como apresentado no capítulo 4, 25,66% da complexidade da codificação dos mapas de profundidade, enquanto é selecionado como melhor modo de predição apenas 1,22% dos casos, entretanto responsável por uma significativa melhora na qualidade do vídeo.

A Figura 26 apresenta a codificação de mapas de profundidade do padrão 3D-HEVC (caixas azuis) integrado ao esquema proposto para redução de complexidade (caixas verdes). No DMMFast, os modos intra tradicional são aplicados sempre, utilizando os algoritmos RMD e MPM (apresentados em 2.2) sem nenhuma modificação. A razão disto é relacionada com o número de vezes que o intra tradicional é selecionado como melhor modo de predição, ou seja, em 98,78% dos blocos, estes na grande maioria dos casos representam regiões homogêneas. Isto ocorre porque os mapas de profundidades possuem muito mais regiões homogêneas do que bordas e em alguns casos de bordas existem direções intra tradicionais capazes de fazer uma boa predição. Após a predição intra tradicional, o algoritmo *Simplified Edge Detector* (SED), detalhado na seção 5.3, é aplicado para classificar o bloco em região constante ou em aresta. Esta classificação é utilizada

para determinar se os novos modos DMM e *Chain Coding* devem ser aplicados ou podem ser desconsiderados para este bloco.

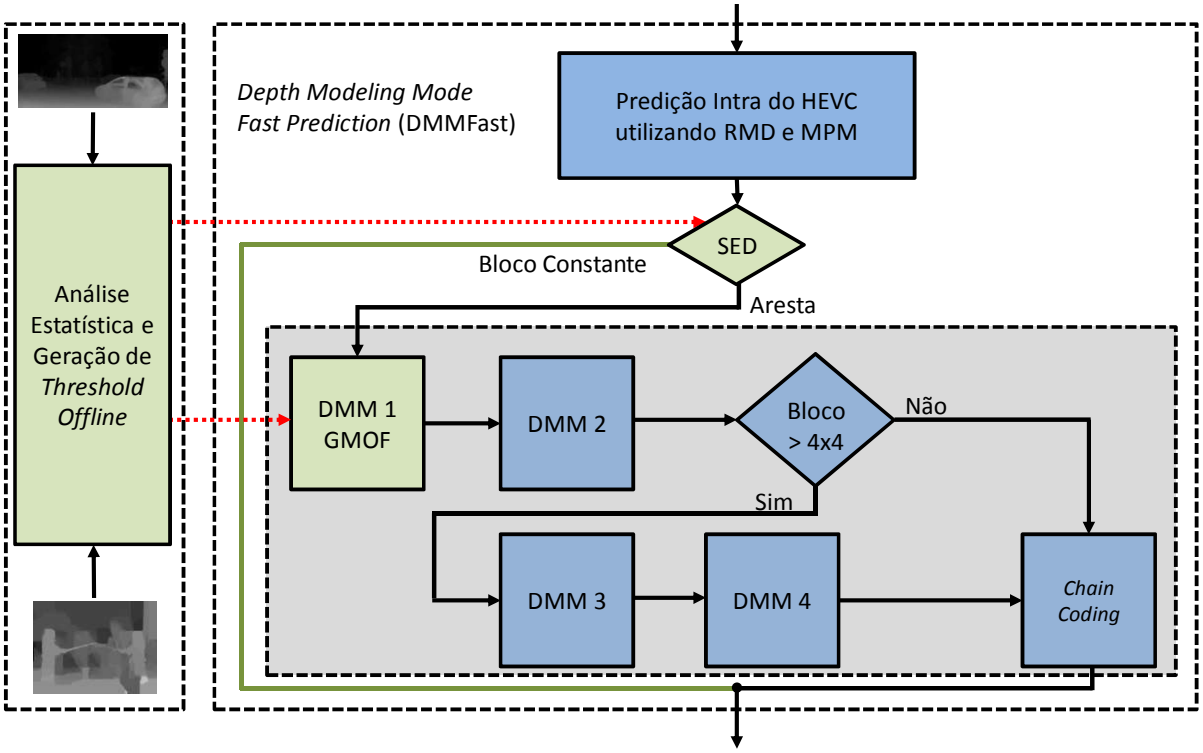



**Figura 26 – Diagrama de blocos do DMMFast.**

O objetivo principal do algoritmo SED é detectar se um bloco é praticamente constante ou se apresenta uma aresta. Como os modos DMM foram especialmente desenvolvidos para preservar as informações de arestas, e apenas poucos blocos em uma cena são arestas, o algoritmo SED possui um alto potencial em termos de redução da complexidade, tal como demonstrado na subseção 5.1.1.

Além disso, o DMMFast é capaz de reduzir a complexidade do DMM 1 através da aplicação do algoritmo *Gradient-based Mode One Filter* (GMOF), apresentado na seção 5.4. O algoritmo GMOF é capaz de reduzir o número de *Wedgelets* avaliadas no processo de DMM 1 ao aplicar um filtro de gradientes nas bordas do bloco e verificando as *N* melhores posições para iniciar ou terminar uma *Wedgelet*.

Combinando as duas soluções, o DMMFast é capaz de eliminar cálculos desnecessários para reduzir a complexidade da codificação dos mapas de profundidade, apresentando pequenas (ou nulas) degradações na eficiência da codificação de vídeo, como será apresentado no capítulo 6.

## 5.3 *Simplified Edge Detector* (SED)

Baseado na análise qualitativa apresentada na subseção 5.1.1, o algoritmo *Simplified Edge Detector* (SED) foi proposto para simplificar os novos modos específicos para a codificação da predição intra de mapas de profundidade do 3D-HEVC. Como os modos DMM e *Chain Coding* possuem uma probabilidade muito baixa de serem escolhidos (1,22% das vezes), caso seja possível encontrar estes casos, para todos os demais esta avaliação pode ser desconsiderada. Assim, quando a predição intra tradicional do HEVC apresenta uma probabilidade dominante (regiões homogêneas), DMMs e *Chain Coding* não necessitam ser computados, pois apresentam um pequeno (ou até nenhum) impacto na eficiência da predição.

O diagrama de blocos do algoritmo SED é apresentado na Figura 27. O algoritmo SED compara o $D_{max}$ (obtido através da equação (4)) com um limiar, chamado aqui de TH, para classificar se o bloco é uma aresta ou uma região homogênea. Esta classificação é feita de acordo com a equação (15). No caso em que $D_{max}$ é maior do que TH, então o bloco é classificado como uma aresta. Caso $D_{max}$ seja menor ou igual, então o bloco é classificado como uma região constante.

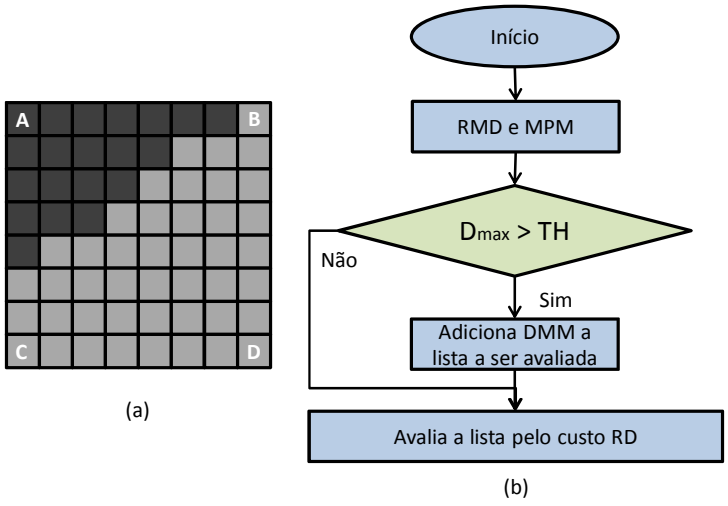



**Figura 27 – (a) Bloco 8x8 (b) Diagrama de blocos do SED.**

$$\text{Classificação} = \begin{cases} \text{Aresta}, & \text{D}_{\max} > TH \\ \text{Constante}, & \text{D}_{\max} \leq \text{TH} \end{cases} \quad (15)$$

Caso o bloco seja classificado como uma aresta, então nenhuma simplificação é feita e então, o fluxo de codificação dos mapas de profundidade do 3D-HEVC tradicional não é modificado. Entretanto, quando o bloco codificado é

classificado como uma região constante, o algoritmo SED não adiciona os novos modos DMMs e *Chain Coding* no processo de predição. Neste caso, apenas os modos de predição intra tradicional do HEVC são avaliados.

Por um lado, a solução proposta aqui é extremamente simples, intuitiva e com um custo computacional baixo. Por outro lado, como será demonstrado nos resultados (capítulo 6) esta solução se provou eficiente. O grande desafio neste caso é a definição dos limiares com a finalidade de garantir um bom compromisso entre a eficiência da codificação e a redução da complexidade. A subseção a seguir apresenta a análise estatística dos modos de codificação em relação ao $D_{max}$, usado para determinar o limiar TH do algoritmo SED.

### 5.3.1 Análise do *Threshold*

Para classificar os blocos com o algoritmo SED, uma análise estatística foi feita utilizando os vídeos *Kendo* e *Poznan_Street*, presentes nas Condições Comuns de Teste (CCT) (RUSANOVSKYY, 2013). Nesta avaliação foram utilizados apenas dois vídeos porque as CCT definem apenas sete vídeos 3D com mapas de profundidade associados. Além disso, é muito difícil encontrar outros vídeos 3D disponíveis para fazer essa avaliação. Dessa forma, optamos por utilizar apenas dois vídeos para a definição do ponto de operação. Com isso é possível separar o conjunto utilizado para geração do ponto de operação do conjunto de teste, tornando os resultados de avaliação mais confiáveis. Estes vídeos foram simulados para os parâmetros de quantização (QP) 25, 30, 35 e 40. Na análise proposta, os blocos codificados foram divididos de acordo com a resolução do vídeo e o tamanho de bloco. Esta divisão é importante porque diferentes tamanhos de blocos e diferentes resoluções podem apresentar diferentes valores ótimos para o limiar TH.

Para cada bloco de profundidade codificado, foi armazenada a informação do melhor modo intra escolhido: DMMs ou predição intra tradicional do HEVC. Além disso, o valor de $D_{max}$ também foi armazenado. A Figura 28 apresenta a distribuição de probabilidades, assumindo uma distribuição Gaussiana observada nos experimentos propostos. Estas curvas foram obtidas na análise para as seguintes condições: (a) e (b) tamanho de bloco 4x4, (c) e (d) tamanho de bloco 8x8, (e) e (f) tamanho de bloco 16x16 e, (g) e (h) tamanho de bloco 32x32. As condições das letras (a), (c), (e), (g) apresentam vídeos de resolução HD 1080p, enquanto nas letras (b), (d), (f) e (h) apresentam vídeos de resolução 1024x768. O $D_{max}$ pode variar

entre 0 a 255, entretanto valores acima de 150 foram omitidos na Figura 28 para apresentar uma maior legibilidade, e também, porque eles apresentam valores inexpressivos de densidade probabilística.

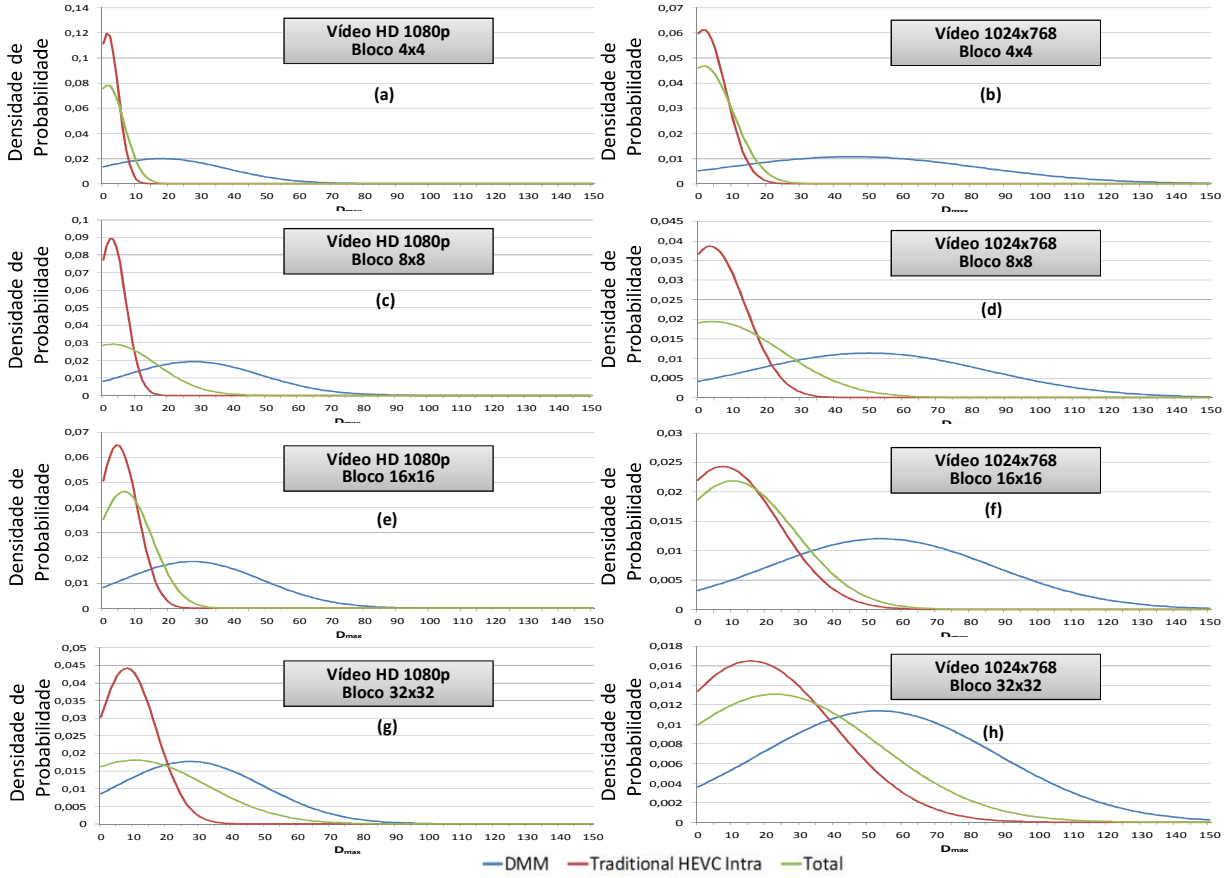



Figura 28 – Análise estatística para o algoritmo SED.

A curva em azul e a curva em vermelho apresentam a densidade de probabilidade para DMMs e intra tradicional, respectivamente. É oportuno notar que estas curvas não estão normalizadas e, por essa razão, a curva em verde apresenta a distribuição de todos os blocos codificados. A análise desta curva é importante, pois o número de seleções dos modos DMMs apresenta um impacto insignificante no total, quando os modos intra tradicionais são considerados.

Analisando a Figura 28 (a) e (b) (blocos 4x4), é possível perceber que a predição intra tradicional do HEVC foi escolhida como a melhor predição para baixos valores de $D_{max}$ (abaixo de 20 e 30, respectivamente). Para maiores valores de $D_{max}$, os modos DMMs se tornaram uma melhor opção. Para a Figura 28 (g) e (h) (blocos 32x32) a probabilidade do intra tradicional é apenas inexpressiva para valores de $D_{max}$ superiores a 50 e 100 para vídeos 1080p e 1024x768, respectivamente. Como

esperado, o limiar TH deve crescer com o aumento do tamanho do bloco. É oportuno ressaltar que para blocos maiores, e sem arestas, se um gradiente suave existir, o $D_{max}$ irá apresentar maiores valores do que comparado com blocos menores, logo, o TH de blocos maiores será maior do que o de blocos menores.

A diferença entre as resoluções também merecem uma consideração especial. É importante neste caso notar que um bloco de tamanho fixo em uma dada resolução apresenta uma menor representatividade em uma resolução maior. Assim, o algoritmo proposto deve considerar diferentes limiares TH de acordo com a resolução do vídeo e o tamanho de bloco.

Através desta análise estatística sobre a seleção dos modos de acordo com o parâmetro $D_{max}$, é possível demonstrar que os modos DMMs são selecionados de forma importante apenas para elevados valores de $D_{max}$, tal como motivado em 5.1.1. Isto ocorre porque valores elevados de $D_{max}$ representam blocos contendo arestas, o que é o caso especial para o qual os modos DMMs foram criados, e onde eles tendem a obter melhores resultados do que os modos intra tradicional.

Os limiares utilizados neste algoritmo foram definidos utilizando resultados experimentais (Figura 28) e a equação (16), onde a média obtida nas curvas do $D_{max}$, foi adicionado a *k* desvios padrão. Através de análises experimentais, *k* foi escolhido como 1 para vídeos HD 1080p e 1,5 para vídeos 1024x768. A média e o desvio padrão foram considerados de acordo com a curva de densidade de probabilidade obtida na análise estatística. Os limiares (TH) utilizados para os diferentes tamanhos de blocos e resoluções são mostrados na Tabela 3.

$$TH = \mu + k\,\sigma \tag{16}$$

Tabela 3 – Limiares utilizados pelo SED.

| Tamanho de bloco / Resolução | 4x4 | 8x8 | 16x16 | 32x32 |
|---|---|---|---|---|
| 1024x768 | 12 | 20 | 34 | 55 |
| 1080p | 8 | 11 | 16 | 25 |

## 5.4 *Gradient-Based Mode One Filter* (GMOF)

O algoritmo GMOF foi proposto para reduzir o número das *Wedgelets* avaliadas pelo algoritmo DMM 1. O algoritmo GMOF é baseado nas observações apresentadas em 5.1.2, que concluíram que os maiores gradientes das bordas do

bloco estão correlacionados com os pontos iniciais das melhores *Wedgelets* e, visando reduzir a complexidade do modo DMM 1, que é o mais complexo dentre os novos modos presentes no 3D-HEVC.

O algoritmo GMOF inicia o seu processo criando uma lista vazia de gradientes no início da execução do DMM 1. Os gradientes da linha superior, da coluna esquerda, da linha inferior e da coluna direita são computados utilizando as equações (11), (12), (13) e (14) e são inseridos na lista de gradientes. A lista proposta está sempre ordenada na ordem decrescente do valor do gradiente.

Após essa avaliação, as primeiras *N* posições nesta lista contém os maiores gradientes. A ideia central é que estes maiores gradientes guiem o processo para as melhores possibilidades de iniciar (ou terminar) uma *Wedgelet*. Após, apenas as *Wedgelets* que iniciem ou terminem nas *N* primeiras posições da lista de gradientes são avaliadas pelo DMM 1. Este processo exclui a maioria das *Wedgelets* do processo de avaliação que seriam convencionalmente avaliadas pelo DMM 1.

O algoritmo proposto é afetado diretamente pelo valor do parâmetro *N*. Quanto menor o valor de *N*, maior será a redução de complexidade esperada pela simplificação do DMM 1. Entretanto, este comportamento não pode ser garantido para o codificador completo, apenas especificamente para o DMM 1. O modo de decisão presente no HTM apresenta um processo recursivo com terminação rápida. Assim, a utilização de um valor maior para *N* pode fazer com que a recursão termine antes, resultando em uma redução de complexidade, o que pode parecer incoerente ao avaliarmos apenas a complexidade do DMM 1. O mesmo tipo de discussão pode ser ampliado em termos de eficiência da codificação. Com isso, para determinar o melhor parâmetro *N* é necessário uma análise experimental, variando o valor do parâmetro e analisando os resultados para diversas configurações.

### 5.4.1 Análise do Valor do Parâmetro *N*

Para medir o impacto do parâmetro *N* na execução do algoritmo GMOF, foram feitas simulações utilizando as sequências *Kendo* e *Poznan_Street* com os seguintes parâmetros *N*: 1, 2, 4, 8 e 16. A Figura 29 (a), (b) e (c) apresentam a redução de complexidade do codificador completo, a redução de complexidade considerando apenas mapas de profundidade e BD-rate (*Bjøntegaard Delta Rate*) (BJONTEGAARD, 2001) de acordo com a variação de *N*, respectivamente.

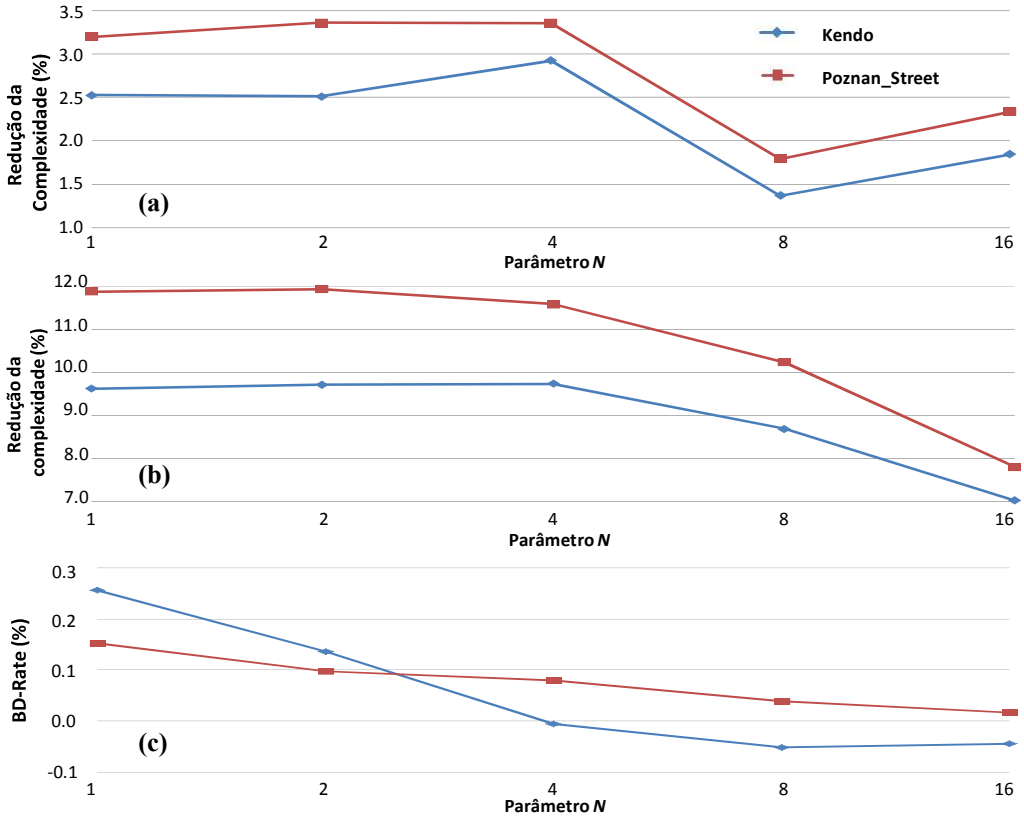



Figura 29 – Analise do algoritmo GMOF de acordo com o parâmetro *N*. (a) Redução de Complexidade no codificador completo. (b) Redução de complexidade considerando apenas mapas de profundidade e (c) Resultados de BD-rate.

Analisando a Figura 29 (a), é possível perceber que a complexidade do codificador completo não é afetada monotonicamente pelo parâmetro *N*. O ponto de menor redução de complexidade é obtido com *N*=8. Isto acontece porque a decisão local de escolher a melhor *Wedgelet* no 3D-HEVC pode resultar um incremento na complexidade nos próximos passos do codificador, incluindo a codificação de textura das próximas vistas. Considerando apenas a codificação de profundidade na Figura 29 (b), para *N* entre 1 e 4, a redução de complexidade é praticamente constante. Se *N* for incrementado ainda mais, a redução de complexidade diminui.

Considerando apenas a complexidade, o melhor ponto de operação (maior redução de complexidade) estaria localizado entre os pontos 1 e 4, com pequenas variações de acordo com o vídeo. Entretanto, analisando o BD-rate (quanto menor melhor) apresentado na Figura 29 (c), pode-se notar que elevando o valor de *N* até 8 é possível reduzir o BD-rate de forma monotônica, porém, aumentando até 16, o BD-rate começa a crescer. Em outras palavras, o melhor ponto de operação em termos de BD-rate é *N*=8.

Através desta avaliação, e focando no objetivo de reduzir a complexidade mantendo a eficiência da codificação, *N*=8 foi escolhido como o melhor ponto de operação, já que, na média, este ponto apresenta um BD-rate de 0,009% (aumento de *bit rate* para a mesma qualidade) com uma significativa redução da complexidade para mapas de profundidade de 9,4%.

Este capítulo apresentou a análise desenvolvida para o desenvolvimento do DMMFast, assim como os algoritmos que lhe compõem: o SED e o GMOF. Além disso, foi apresentado um estudo detalhado, mostrando a definição dos pontos de operação ideais para estes algoritmos, considerando dois vídeos das Condições Comuns de Teste (CCT) nesta análise. O próximo capítulo apresenta os resultados objetivos e subjetivos de qualidade destes algoritmos, além de apresentar uma comparação com trabalhos relacionados.

# 6 CENÁRIO DE AVALIAÇÕES E RESULTADOS

Este capítulo apresenta inicialmente as configurações utilizadas para realizar as simulações, e posteriormente, os resultados alcançados pelos algoritmos propostos através de avaliação objetiva com simulações em software e também avaliações subjetivas, feitas por meio da análise dos vídeos codificados utilizando o DMMFast e reproduzidos em um televisor.

## 6.1 Cenário de avaliações - Condições Comuns de Teste

As Condições Comuns de Teste (CCT) (RUSANOVSKYY, 2013) foram desenvolvidas para permitir uma comparação justa entre trabalhos relacionados. Elas englobam características significativas na codificação de vídeos 3D, e utilizam diversas sequências de teste com diferentes características. Para avaliar um novo algoritmo no padrão emergente 3D-HEVC, elas devem ser utilizadas.

No 3D-HEVC são codificadas as seguintes sequências de teste para as CCTs: *Poznan_Hall2*, *Poznan_Street*, *Undo_Dancer*, *GT_Fly*, *Kendo*, *Balloons* e *Newspaper1*. Estas sequências serão apresentadas mais adiante, na subseção 6.1.1.

Existem dois cenários a serem considerados: o primeiro para sequências de teste com dados de profundidade, e o segundo, para sequências de teste sem dados de profundidade. Em ambos os cenários, são analisados os casos de codificação com duas e três vistas. Neste trabalho, será considerado apenas o caso com dados de profundidade utilizando três vistas, uma vez que as soluções propostas visam à redução de complexidade na codificação de mapas de profundidade.

Todos os vídeos utilizados na codificação são compostos por várias vistas. A Tabela 4 mostra as vistas utilizadas pelas CCTs para codificação com duas e com três vistas. Além disto, na Tabela 4 são mostradas as resoluções de cada vídeo e a taxa de amostragem (em quadros por segundo).

Os valores de QP utilizados nas simulações são definidos pelas CCTs. As simulações da vista base devem ser realizadas com parâmetros de QP nos valores: 25, 30, 35 e 40. Os valores do QP das vistas de profundidade são dados pela Tabela 5, onde $QP_V$ significa o QP utilizado para codificar o quadro de textura e $QP_D$ significa o QP utilizado para codificar o quadro de profundidade associado a este quadro de textura.

Tabela 4 – Vistas utilizadas pelas CCTs para codificação com duas ou três câmeras.

| Seq. ID | Sequência de Teste | Resolução | Taxa de amostragem | 2 vistas | 3 vistas | Número de Câmeras Filmadas |
|---|---|---|---|---|---|---|
| **S01** | *Poznan_Hall2* | 1920x1080 | 25 | 7-6 | 7-6-5 | 9 (1D – 13,5 cm) |
| **S02** | *Poznan_Street* | 1920x1080 | 25 | 5-4 | 5-4-3 | 9 (1D – 13,5 cm) |
| **S03** | *Undo_Dancer* | 1920x1080 | 25 | 1-5 | 1-5-9 | 9 (1D – 13,5 cm) |
| **S04** | *GT_Fly* | 1920x1080 | 25 | 9-5 | 9-5-1 | 9 (1D – 13,5 cm) |
| **S05** | *Kendo* | 1024x768 | 30 | 1-3 | 1-3-5 | 7 (1D – 5 cm) |
| **S06** | *Balloons* | 1024x768 | 30 | 1-3 | 1-3-5 | 7 (1D – 5 cm) |
| **S08** | *Newspaper1* | 1024x768 | 30 | 2-4 | 2-4-6 | 9 (1D – 13,5 cm) |

Tabela 5 – Valores do QP de profundidade em função do QP 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 |

### 6.1.1 Sequências de teste utilizadas nas CCTs

Esta subseção apresenta as principais características de cada uma das sequências de testes definidas pelas CCTs.

As sequências *Poznan_hall2*, *Poznan_Street*, *Undo_Dancer* e *GT_Fly* possuem resolução de 1920x1080 pixels, e foram filmadas com uma taxa de amostragem de 25 quadros por segundo. Estas quatro sequências foram capturadas com nove câmeras com um espaçamento de 13,75 cm entre elas.

A sequência *Poznan_Hall2* foi filmada dentro da *Poznan University of Technology*. Suas características são: ambiente interno, movimentação de objetos complexas (o que inclui reflexões e transparência), câmera em movimento, nível de detalhes médio e uma estrutura complexa de profundidade. Na Figura 30 é mostrada uma imagem da sequência *Poznan_Hall2*, onde é possível identificar algumas dessas características. Na Figura 31 é mostrada uma imagem de profundidade da câmera relacionada com a imagem apresentada na Figura 30.

**Figura 30 – Imagem de textura da sequência *Poznan_Hall2*.**

**Figura 31 – Imagem de profundidade da sequência *Poznan_Hall2*.**

A sequência *Poznan_street* possui as seguintes características: ambiente externo, movimentação de objetos complexas (o que inclui reflexões e transparência), câmera estática, alto nível de detalhes, uma estrutura complexa de profundidade e luz natural. A Figura 32 e a Figura 33 mostram uma imagem da sequência *Poznan_street* e seu mapa de profundidade, respectivamente.

**Figura 32 – Imagem de textura da sequência *Poznan_street*.**

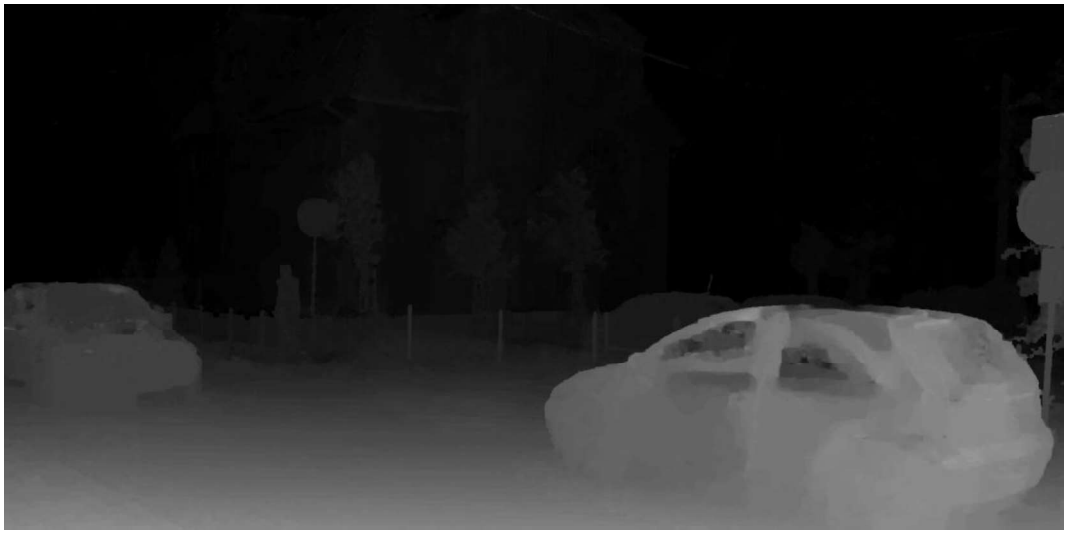


**Figura 33 – Imagem de profundidade da sequência *Poznan_street*.**

A sequência *Undo_Dancer* apresenta um vídeo sintético, e foi produzida utilizando computação gráfica. Nela não estão presentes movimentos complexos. A câmera é móvel e o vídeo possui um alto nível de detalhes. A estrutura de profundidade presente nessa sequência é simples. A Figura 34 e a Figura 35 apresentam uma imagem da sequência *Undo_Dancer* e seu mapa de profundidade, respectivamente.

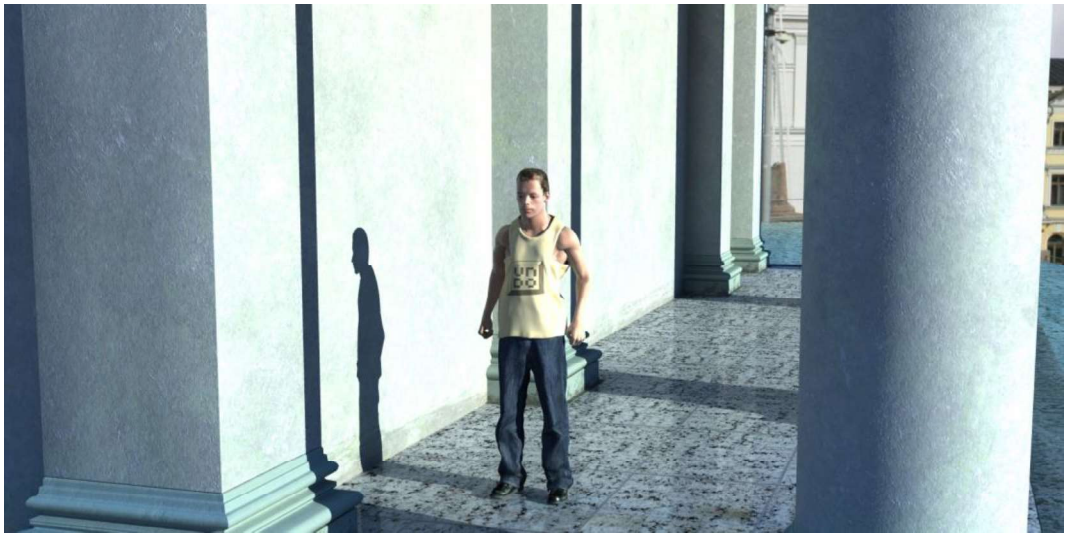


**Figura 34 – Imagem de textura da sequência *Undo_Dancer*.**

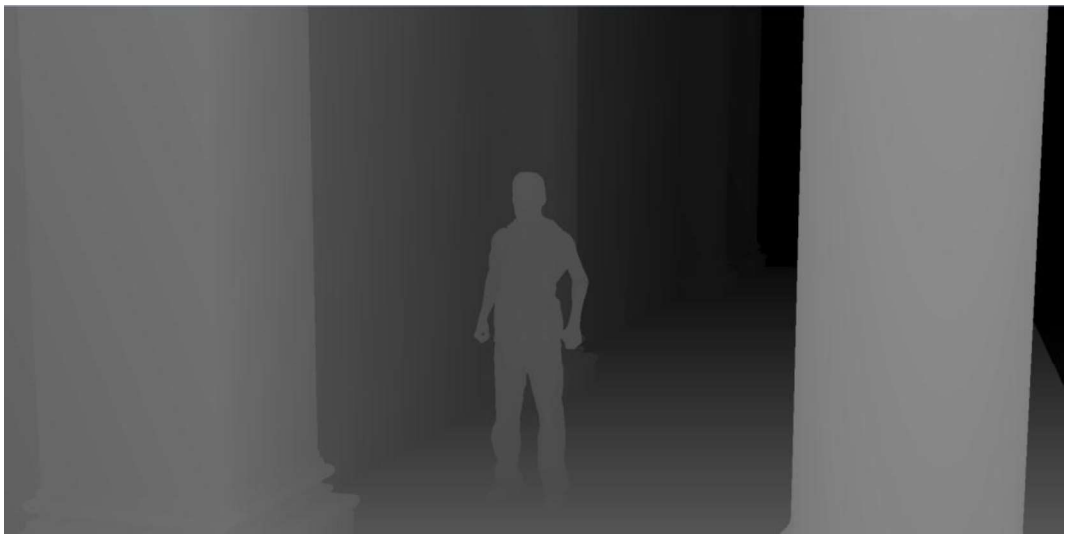


**Figura 35 – Imagem de profundidade sequência *Undo_Dancer*.**

A sequência *GT_Fly* também foi criada utilizando computação gráfica. Nesta sequência as seguintes características estão presentes: simples movimentação de objetos, médio nível de detalhes, estrutura de profundidade complexa. Na Figura 36 e na Figura 37 são mostradas a imagem inicial da sequência *GT_Fly* e seu mapa de profundidade, respectivamente.

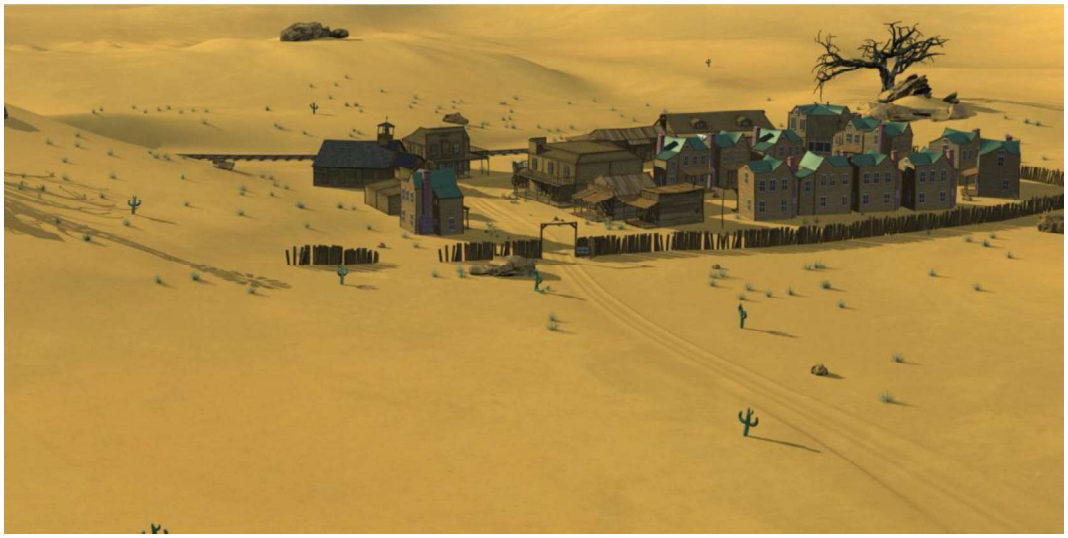


**Figura 36 – Imagem de textura da sequência *GT_Fly*.**

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

As sequências *Kendo*, *Balloons* e *Newspaper1* possuem uma resolução de 1024x768 pixels. As três foram filmadas a uma taxa de amostragem de 30 quadros por segundo. As sequências *Kendo* e *Balloons* foram filmadas pelo *Tanimoto Laboratory na Nagoya University* com sete câmeras com um espaçamento de 5 cm entre elas enquanto que a sequência *Newspaper1* foi filmada no *Gwangju Institute of Science and Technology* (GIST) com 9 câmeras com um espaçamento de 13,75 cm entre elas.

As características da sequência *Kendo* são: ambiente interno, feito em estúdio, movimentações de objetos complexas (incluindo fumaça, reflexão e transparência), a câmera se move, alto nível de detalhes, estrutura de profundidade

com complexidade média e luz de estúdio. A Figura 38 e a Figura 39 apresentam uma imagem da sequência *Kendo* e seu mapa de profundidade, respectivamente.

Na sequência *Balloons* estão presentes as seguintes características: ambiente interno, feito em estúdio, alta complexidade de movimento (incluindo reflexões e transparência), a câmera se move, nível médio de detalhes, nível médio da complexidade da estrutura de profundidade e luz de estúdio. A Figura 40 e a Figura 41 apresentam uma imagem da sequência *Balloons* e seu mapa de profundidade, respectivamente.

**Figura 38 – Imagem de textura da sequência *Kendo*.**

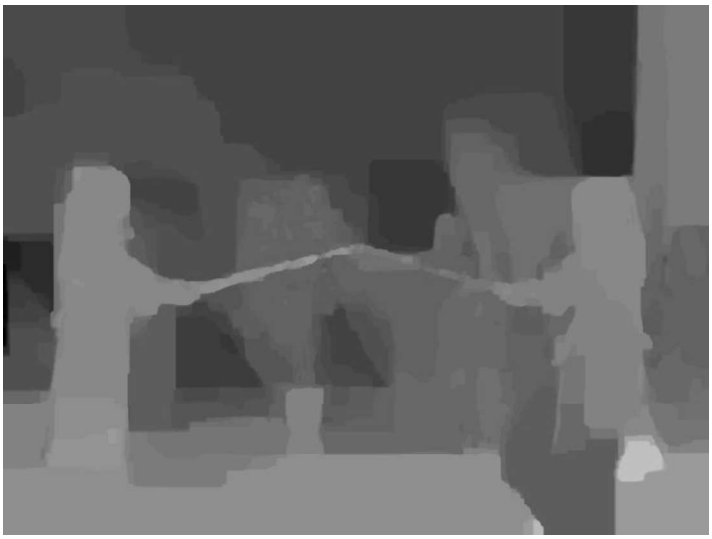


**Figura 39 – Imagem de profundidade da sequência *Kendo*.**

**Figura 40 – Imagem de textura da sequência *Balloons*.**

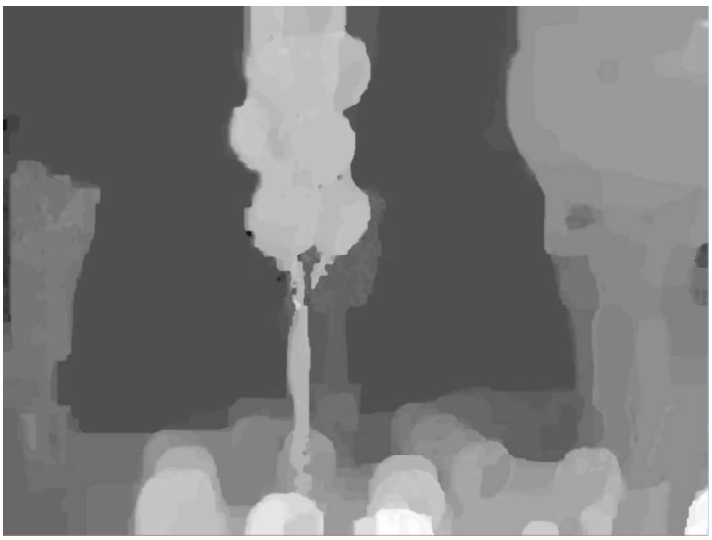


**Figura 41 – Imagem de profundidade da sequência *Balloons*.**

As características da sequência *Newspaper1* são: Ambiente interno, feito em estúdio, movimentos de objetos simples, câmera estática, alto nível de detalhes, estrutura de profundidade com complexidade média e luz de estúdio. A Figura 42 e a Figura 43 apresentam uma imagem da sequência *Newspaper1* e seu mapa de profundidade. É oportuno ressaltar que o mapa de profundidade do vídeo *Newspaper1* apresenta diversas falhas por problemas na filmagem como pode-se notar um ponto preto no rosto da mulher à direita na Figura 43.

**Figura 42 – Imagem de textura da sequência *Newspaper1*.**

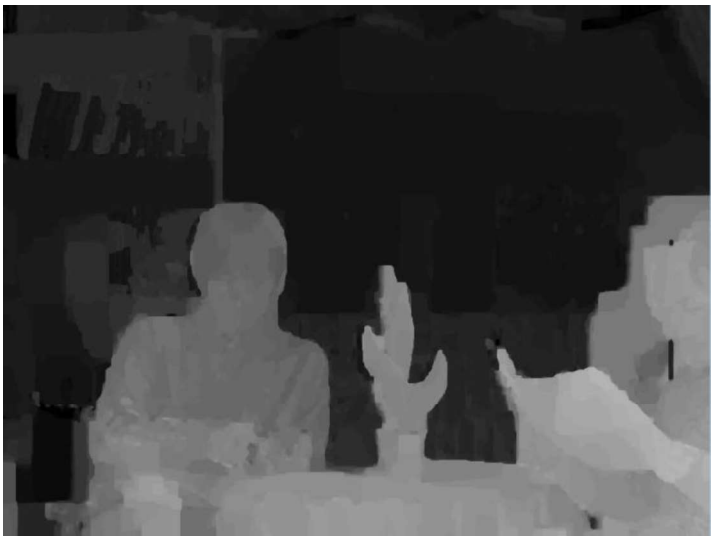


**Figura 43 – Imagem de profundidade da sequência *Newspaper1*.**

A Tabela 6 apresenta um resumo das características dessas sequências, conforme apresentados anteriormente. Algumas destas características foram obtidas em (MOBILE-3DTV, 2013). As características que não estavam disponíveis em (MOBILIE-3DTV, 2013) foram obtidas através da análise visual das sequências.

## 6.2 Avaliações em Software

Esta seção apresenta os resultados da avaliação objetiva dos algoritmos SED, GMOF e do esquema DMMFast. Os algoritmos propostos foram avaliados no HTM-7.0, utilizando as Condições Comuns de Teste (CCT) (RUSANOVSKYY, 2013).

Os dados originais, utilizados para realizar a comparação com o nossos algoritmos, foram gerados a partir da codificação das sequências de teste presentes nas CCT, utilizando o software HTM-7.0 sem nenhuma modificação. Os algoritmos

Tabela 6 – Características das sequências de teste utilizadas pelas CCTs.

| Sequência de Teste | Resolução | Amost. | Ambiente | Movimentos de objetos | Câmera | Nível de detalhes | Estrutura de profundidade |
|---|---|---|---|---|---|---|---|
| ***Poznan_Hall2*** | 1920x1080 | 25 | Interno | Complexos (reflexões e transparência) | Móvel | Médio | Complexa |
| ***Poznan_street*** | 1920x1080 | 25 | Externo | Complexos (reflexões e transparência) | Estática | Alto | Complexa |
| ***Undo_Dancer*** | 1920x1080 | 25 | Comp. Gráfica | Simples | Móvel | Alto | Simples |
| ***GT_Fly*** | 1920x1080 | 25 | Comp. Gráfica | Simples | Móvel | Médio | Complexa |
| ***Kendo*** | 1024x768 | 30 | Interno Estúdio | Complexos (fumaça, reflexão e transparência) | Móvel | Alto | Média |
| ***Balloons*** | 1024x768 | 30 | Interno Estúdio | Complexas (reflexões e transparência) | Móvel | Médio | Média |
| ***Newspaper1*** | 1024x768 | 30 | Interno Estúdio | Simples | Estática | Alto | Média |

desenvolvidos foram inseridos no HTM-7.0 e então executado novamente sobre as sequências de teste das CCT. Os resultados obtidos com a aplicação do DMMFast foram comparados com os resultados originais, utilizando a métrica BD-rate (BJONTEGAARD, 2001). Esta métrica faz uma relação entre a taxa de bits e a qualidade objetiva do vídeo codificado. Nesta dissertação os resultados de BD-rate serão mostrados em valores percentuais, onde valores positivos indicam um incremento na taxa de bits para uma qualidade de vídeo equivalente.

Nos experimentos realizados neste trabalho foi considerada apenas a configuração de 3 vistas, previsto nas CCT, onde para cada sequência de vídeo existem 3 canais de textura e 3 canais de profundidade a serem codificados. Os canais de textura serão chamados de vídeo 0 (vista base, que é a vista central da codificação), 1 e 2 no resto desta dissertação.

Em todas as avaliações, a codificação do vídeo 0 (textura) não foi modificada porque esta não apresenta nenhuma dependência com a codificação de mapas de profundidade. A codificação dos vídeos 1 e 2 apresentou variações desprezíveis, apesar da codificação destas texturas apresentar dependências com todas as vistas já codificadas, inclusive com mapas de profundidades por predição *inter-component*. Os resultados mais importantes deste trabalho estão relacionados com as vistas sintetizadas, isto porque o DMMFast é aplicado apenas na codificação de mapas de profundidade, e os mapas de profundidade refletem diretamente na qualidade das

vistas sintetizadas, já que não são visualizados de forma independe. A síntese das vistas foi feita utilizando o algoritmo *View Synthesis Reference Software* (VSRS) versão 3.5, descrito em (MULLER, 2008), e disponível pelo *Joint Collaborative Team on 3D Video Coding Extension Development* (JCT-3V) (JCT-3V, 2014).

Para o cálculo do BD-rate das vistas sintetizadas, foi considerado o PSNR de todas as vistas sintetizadas (nove) e a soma dos *bit rates* de todas as vistas de textura e mapas de profundidade, tal como a planilha disponibilizada em (RUSANOVSKYY, 2013).

Além do exposto acima, nesta seção são apresentadas comparações com o trabalho (GU, 2013b), o único encontrado na literatura com resultados de redução de complexidade para o 3D-HEVC que ainda não foi inserido no HTM-7.0.

### 6.2.1 *Depth Modeling Modes Fast Prediction* (DMMFast)

A Tabela 7 apresenta os resultados do codificador 3D-HEVC contendo o DMMFast. Neste cenário, é possível obter uma redução de complexidade para a codificação de profundidade, em média, de 24,5%.

Tabela 7 – Resultados do DMMFast (CCT).

| Vídeos | vídeo 1 (BD-rate) | vídeo 2 (BD-rate) | Todos vídeos (BD-rate) | Vistas Sintetizadas (BD-rate) | Redução de Complexidade (tempo de codificação) | |
|---|---|---|---|---|---|---|
| | | | | | Completo | Profundidade |
| *Balloons* | 0,130% | -0,106% | 0,031% | 0,345% | 5,8% | 21,7% |
| *Kendo* | -0,049% | 0,054% | 0,015% | 0,145% | 5,7% | 20,8% |
| *Newspaper_CC* | -0,157% | -0,072% | -0,044% | -2,540% | 6,5% | 23,8% |
| *GT_Fly* | 0,256% | -0,380% | 0,027% | 0,156% | 6,2% | 24,6% |
| *Poznan_Hall2* | -0,002% | 0,576% | 0,123% | 0,547% | 6,3% | 26,9% |
| *Poznan_Street* | 0,037% | -0,272% | -0,016% | 0,272% | 7,4% | 27,2% |
| *Undo_Dancer* | 0,171% | 0,045% | 0,036% | 0,470% | 6,0% | 26,2% |
| 1024x768 | -0,026% | -0,041% | 0,001% | -0,683% | 6,0% | 22,1% |
| 1920x1088 | 0,116% | -0,008% | 0,043% | 0,361% | 6,5% | 26,2% |
| **Média** | **0,055%** | **-0,022%** | **0,025%** | **-0,086%** | **6,3%** | **24,5%** |

Considerando o codificador completo, é possível obter uma redução de complexidade de 6,3%. Considerando apenas as vistas sintetizadas, o DMMFast obtém um ganho de *bit rate* de 0,086%, principalmente devido ao ganho de 2,54% em termos de *bit rate* obtidos pela sequência *Newspaper_CC*. Este ganho de *bit rate* é explicado devido aos ganhos obtidos pelos algoritmos SED e GMOF que serão descritos em 6.2.2 e 6.2.3.

Removendo o vídeo *Newspaper_CC* dos resultados médios, o DMMFast implica em um aumento de 0,323% em *bit rate*. Este resultado é uma adição

aceitável já que os ganhos em termos de redução de complexidade atingem 6,3% da complexidade do codificador completo.

O DMMFast também foi avaliado considerando a estrutura de codificação *All-Intra* (AI), caso em que todos os quadros são codificados apenas pela predição intra quadros e não está previsto diretamente nas CCT, mas é importante para a avaliação deste trabalho. No codificador AI 3D-HEVC, a complexidade da codificação de profundidade é superior à complexidade de textura. Isto ocorre porque além de utilizar a predição tradicional de textura, a profundidade ainda adiciona os modos DMM.

A redução de complexidade obtida no codificador AI é apresentada na Figura 44. Na Figura 44 são apresentadas a média, o primeiro, o segundo e o terceiro quartis além do máximo e o mínimo da redução de complexidade de mapas de profundidade para cada uma das sequencias. Pode-se notar na Figura 44 que o DMMFast consegue atingir uma redução da complexidade média acima de 35% para todas as sequencias. Em média, é possível obter 64% de redução da complexidade na codificação de profundidade e 40% para o codificador completo (não mostrado na Figura 44). Essa variação em redução da complexidade mencionada acima ocorre principalmente porque o número de blocos classificados como aresta ou região homogênea varia de quadro para quadro em uma dada sequencia. Os maiores ganhos são obtidos quando a maioria dos blocos é classificada como regiões homogêneas, enquanto os menores ganhos são obtidos quando a maioria é classificada como aresta.

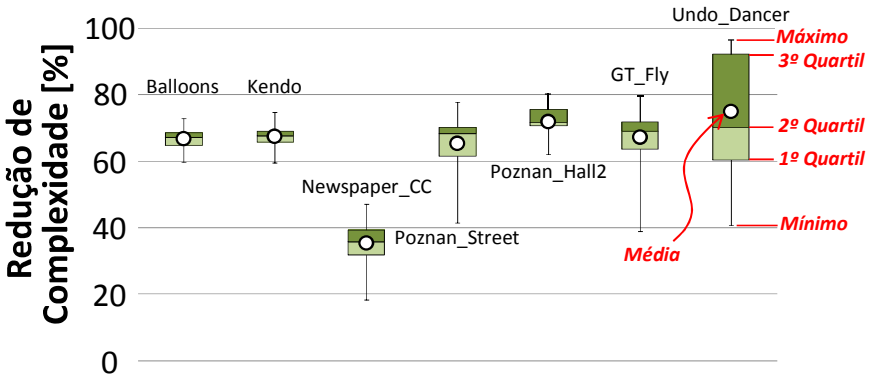



Figura 44 – Redução da complexidade no caso all-intra.

As próximas duas subseções apresentam os resultados obtidos individualmente por cada um dos algoritmos que compõem o DMMFast: o SED e o GMOF.

### 6.2.2 *Simplified Edge Detector* (SED)

A Figura 45 apresenta os percentuais de vezes em que os modos DMM não foram avaliados ao utilizar o algoritmo SED. A Tabela 8 apresenta os resultados da codificação segundo as CCTs, onde os resultados para a primeira vista de textura (vídeo 0) são omitidos, visto que não apresentam nenhuma variação em relação aos resultados originais. O algoritmo SED foi capaz de evitar a avaliação dos DMMs em mais de 90% dos blocos de cada vídeo. Em média, ao utilizar o algoritmo SED, foi possível obter um ganho de *bit rate* de 0,064%. Esse ganho está bastante relacionado aos resultados obtidos para a sequência *Newspaper_CC*, onde foi possível obter uma redução de *bit rate* de 2,062%. Este ganho ocorreu porque o codificador 3D-HEVC toma uma decisão local ao codificar mapas de profundidade, considerando apenas a distorção dos mapas de profundidade preditos e a quantidade de bits necessária para codificar este mapa de profundidade. De fato, o mapa de profundidade não será visualizado pelo usuário final, ele será apenas utilizado para o algoritmo de geração das vistas intermediárias. Como a codificação de profundidade do 3D-HEVC não considera a qualidade da vista sintetizada para a escolha dos melhores modos de predição, é possível reduzir a complexidade do processo e ainda obter um ganho em termos de *bit rate*.

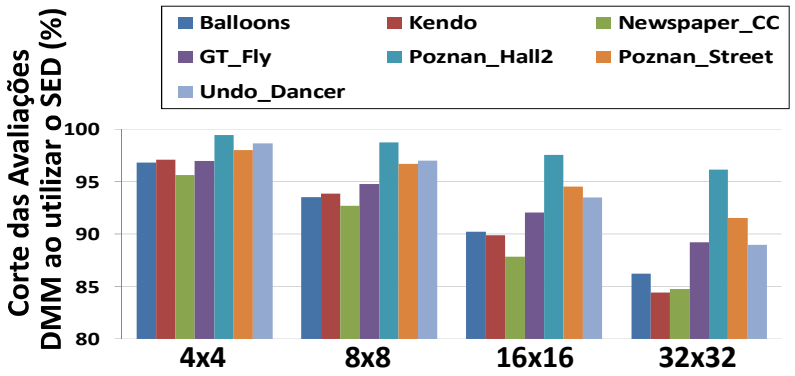



Figura 45 – Porcentagem dos DMM não codificados utilizando o algoritmo SED

O vídeo *Newspaper_CC* possui muitas regiões contendo várias arestas com pequenos valores em $D_{max}$. Quando o algoritmo SED foi aplicado neste vídeo, esses blocos que contem pequenas diferenças em $D_{max}$ foram codificados utilizando a predição intra tradicional do HEVC. Usar a predição intra tradicional neste caso, ajudou a reduzir o *bit rate*, mantendo a mesma qualidade de vídeo. Desconsiderando o vídeo *Newspaper_CC*, em média, utilizar o algoritmo SED provocou um aumento de 0,269% do *bit rate* (para uma mesma qualidade de vídeo).

Tabela 8 – Resultados do Algoritmo SED (CCT).

| Vídeos | vídeo 1 (BD-rate) | vídeo 2 (BD-rate) | Todos vídeos (BD-rate) | Vistas Sintetizadas (BD-rate) | Redução de Complexidade (tempo de codificação) | |
|---|---|---|---|---|---|---|
| | | | | | Completo | Profundidade |
| *Balloons* | 0,107% | 0,427% | 0,095% | 0,282% | 5,4% | 20,8% |
| *Kendo* | -0,025% | -0,169% | -0,020% | 0,153% | 5,1% | 20,1% |
| *Newspaper_CC* | -0,162% | 0,222% | -0,001% | -2,062% | 6,0% | 22,9% |
| *GT_Fly* | 0,172% | -0,216% | 0,029% | 0,173% | 6,0% | 24,1% |
| *Poznan_Hall2* | 0,036% | 0,260% | 0,084% | 0,487% | 6,0% | 26,7% |
| *Poznan_Street* | 0,008% | -0,153% | -0,016% | 0,217% | 7,1% | 26,6% |
| *Undo_Dancer* | 0,220% | -0,123% | 0,015% | 0,303% | 5,7% | 25,3% |
| 1024x768 | -0,027% | 0,160% | 0,025% | -0,542% | 5,5% | 21,3% |
| 1920x1088 | 0,109% | -0,058% | 0,028% | 0,295% | 6,2% | 25,7% |
| **Média** | **0,051%** | **0,036%** | **0,027%** | **-0,064%** | **5,9%** | **23,8%** |

Através da análise da complexidade do algoritmo proposto, em relação ao HTM-7.0 original, foi possível obter um ganho médio de 5,9%, considerando o codificador completo (textura e profundidade). De fato, considerando apenas a codificação do canal de profundidade, foi possível obter uma redução da complexidade de 23,8%. O algoritmo SED provocou uma redução de complexidade significativa, já considerando a inserção de uma complexidade mínima para cálculo de $D_{max}$.

### 6.2.3 *Gradient-Based Mode One Filter* (GMOF)

A Figura 46 apresenta o número médio de *Wedgelets* avaliadas no DMM 1, quando o algoritmo GMOF é aplicado, usando como condições o parâmetro *N* igual a 8 e o QP igual a 25. É possível perceber, a partir da Figura 46, que o algoritmo proposto é capaz de remover mais de 50% das *Wedgelets* avaliadas no DMM 1 para blocos 4x4 e mais de 75% para os outros tamanhos de blocos.

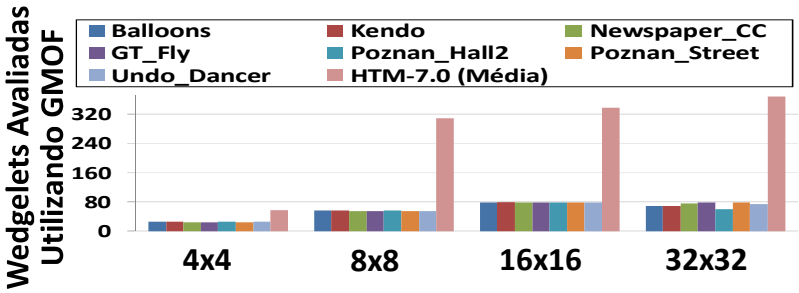



Figura 46 – Número de Wedgelets Avaliadas no DMM 1 usando o algoritmo GMOF.

A Tabela 9 apresenta os resultados do algoritmo GMOF, quando avaliado segundo as CCTs. Em média, o algoritmo GMOF apresenta uma redução de *bit rate*

de 0,047%. Neste algoritmo é possível obter uma redução de complexidade de 1,7%, considerando o codificador completo, e 9,8% considerando apenas a codificação de profundidade.

O melhor resultado em termos de *bit rate* foi obtido na sequência *Undo_Dancer*, que é uma sequência sintética, gerada através de computação gráfica, possui arestas e regiões constantes perfeitamente definidas, o que não acontece em vídeos naturais. Pode-se notar que esta sequencia apresenta mapas de profundidade perfeitos, sem erros de captura, como um vídeo natural (que não foi gerado por computação gráfica) apresentaria. Neste vídeo, foi possível obter um ganho de *bit rate* de 0,577%. O vídeo GT_Fly também apresenta um mapa de profundidade perfeito, gerado através de computação gráfica, porém, apresenta uma perda de BD-rate de 0,139%. Esta perda acontece porque este vídeo apresenta objetos muito pequenos e finos em seus mapas de profundidade, o que dificulta consideravelmente a sua codificação.

Tabela 9 – Resultados do Algoritmo GMOF (CCT).

| Vídeos | vídeo 1 (BD-rate) | vídeo 2 (BD-rate) | Todos vídeos (BD-rate) | Vistas Sintetizadas (BD-rate) | Redução de Complexidade (tempo de codificação) | |
|---|---|---|---|---|---|---|
| | | | | | Completo | Profundidade |
| *Balloons* | 0,112% | 0,023% | 0,028% | 0,045% | 2,8% | 9,5% |
| *Kendo* | -0,035% | -0,114% | -0,007% | -0,021% | 1,3% | 8,6% |
| *Newspaper_CC* | -0,102% | -0,054% | -0,029% | -0,052% | 1,4% | 8,7% |
| *GT_Fly* | 0,132% | 0,191% | 0,040% | 0,139% | 1,6% | 9,4% |
| *Poznan_Hall2* | -0,135% | 0,594% | 0,107% | 0,102% | 1,4% | 11,0% |
| *Poznan_Street* | -0,060% | -0,031% | -0,017% | 0,039% | 1,8% | 10,2% |
| *Undo_Dancer* | 0,079% | 0,213% | 0,031% | -0,577% | 1,8% | 10,8% |
| 1024x768 | -0,008% | -0,048% | -0,003% | -0,010% | 1,8% | 8,9% |
| 1920x1088 | 0,004% | 0,242% | 0,040% | -0,074% | 1,6% | 10,4% |
| **Média** | **-0,001%** | **0,117%** | **0,022%** | **-0,047%** | **1,7%** | **9,8%** |

No HTM-7.0, o conjunto de *Wedgelets* é avaliado apenas para se encontrar a mínima distorção e, assim, tomando uma decisão local (sem considerar o custo RD de todas *Wedgelets*). Esse fato explica o ganho obtido na sequência *Undo_Dancer* (e também em *Kendo* e *Newspaper_CC*), pois ao inserir o algoritmo GMOF no processo do DMM 1, é possível descartar *Wedgelets* que possuem um baixo gradiente nas bordas. No entanto, estas *Wedgelets* poderiam ser selecionadas como a melhor predição no esquema HTM-7.0 tradicional. Quando o algoritmo GMOF é aplicado, todas as *Wedgelets* avaliadas apresentam uma probabilidade considerável

de estar localizadas em uma aresta do bloco, o que contribui para o processo global do codificador.

Os ganhos de complexidade são obtidos porque várias avaliações de *Wedgelets* são descartadas, já que o algoritmo não considera *Wedgelets* que comecem ou terminem de posições que não representem uma aresta, segundo os critérios de avaliação propostos neste trabalho.

Considerando o DMMFast, a adição do algoritmo GMOF para a codificação do DMM 1, em conjunto com a aplicação do algoritmo SED aumenta o ganho de redução de complexidade em 0,7% em relação ao SED, considerando apenas a complexidade da profundidade. A redução de complexidade dos dois algoritmos não é simplesmente somada porque o algoritmo SED consegue reduzir acima de 90% das avaliações dos DMM, e o GMOF é apenas aplicado nos casos em que o DMM 1 continua sendo avaliado.

### 6.2.4 Comparação com trabalhos relacionados

Os únicos trabalhos encontrados na literatura que podem ser comparados de forma justa com os resultados deste trabalho são apresentados em (GU, 2013b) e (ZHANG, 2014). O trabalho de (GU, 2013b) reduz a complexidade da predição intra de mapas de profundidade do 3D-HEVC verificando se o primeiro modo intra a ser avaliado é o modo planar. Se isso for verdade, a técnica apresentada em (GU, 2013b) decide remover todas as avaliações do DMM. Caso seja falso, o fluxo de codificação original do 3D-HEVC também é feito. Além disso, a técnica proposta por (GU, 2013b) calcula a variância do bloco e verifica se está variância é alta. Caso positivo, o algoritmo realiza o fluxo convencional, caso contrário, ele descarta os modos DMM.

O trabalho apresentado em (GU, 2013b) também foi avaliado no HTM-7.0 e por isso a comparação com este trabalho é justa. Na verdade, esta otimização foi escolhida para ser adotada na versão seguinte do software de referência, o HTM-8.0. Este trabalho consegue obter uma redução de complexidade de 2,5% com uma diminuição no BD-rate de 0,01%, quando avaliado segundo as CCT. O DMMFast consegue atingir um ganho 6,3%, além disso, é capaz de obter um ganho de aproximadamente 0,09% no BD-Rate. É necessário mencionar que ambas as técnicas poderiam ser utilizadas em conjunto, podendo obter uma taxa de redução de complexidade ainda maior.

O trabalho de (ZHANG, 2014) foi avaliado no HTM-5.1, na configuração AI, e obtém uma redução de complexidade média de 40,3% para o codificador completo, com perdas de BD-rate de 0,9%. O DMMFast é capaz de reduzir a complexidade em 40%, com perdas desprezíveis de qualidade. Além disso, vale salientar que o algoritmo de (ZHANG, 2014) foi avaliado no HTM-5.1, onde ainda não estavam implementados algoritmos de redução da complexidade como o de (MORA, 2013), entre outros. Neste contexto, existiam possibilidades maiores para a exploração de soluções de redução da complexidade. Além disso, considerando os requisitos de acesso a memória do sistema, o algoritmo proposto por (ZHANG, 2014) necessita de vários acessos à memória, que armazena dados de textura, durante a codificação de profundidade para determinar se os modos DMM devem ser realizados. O DMMFast não possui essa necessidade, o que o torna uma solução mais eficiente neste aspecto.

Estes resultados mostram que os algoritmos propostos neste trabalho, assim como o esquema DMMFast, apresentam resultados significativos, inclusive, superando os resultados obtidos por algoritmos que foram recentemente incorporados ao padrão com a mesma finalidade. Como o 3D-HEVC ainda está em fase de desenvolvimento, os algoritmos desenvolvidos neste trabalho são potenciais candidatos para serem inseridos em uma futura versão do 3D-HEVC. Além disso, os resultados obtidos neste trabalho são importantes para uma implementação eficiente do codificador 3D-HEVC em hardware, focando na codificação em tempo real com menor consumo de energia e recursos de hardware.

## 6.3 Avaliação Subjetiva de Qualidade

As soluções de redução de complexidade propostas neste trabalho visam reduzir a complexidade do processo de codificação de vídeos 3D usando o modelo MVD. Para isso, novas técnicas para reduzir a complexidade de codificação dos mapas de profundidade foram desenvolvidas. Os mapas de profundidade são essenciais no modelo MVD para a geração das vistas sintetizadas, geradas a partir das informações de textura e profundidade de vistas vizinhas. Os resultados apresentados na seção anterior demonstram que o esquema DMMFast apresenta resultados significativos de redução de complexidade, sem inserir perdas significativas na qualidade e na taxa de bits do vídeo codificado. No entanto, nem sempre a qualidade objetiva está diretamente associada à qualidade subjetiva. Ou

seja, mesmo sem inserir perdas significativas na qualidade objetiva, dados os critérios utilizados para esta avaliação (PSNR, *bit rate* e BD-Rate), a qualidade subjetiva (percebida pelos espectadores) pode sofrer uma degradação significativa.

Para avaliar este impacto, uma análise subjetiva da qualidade das vistas sintetizadas produzidas pela codificação usando o DMMFast também foi desenvolvida. Nesta análise foi utilizada a técnica chamada *Degradation Category Rating* (DCR) (ITU, 2009). O DCR define que o experimento deve ser realizado em uma sala contendo uma TV ou um monitor com capacidade para reproduzir vídeos de alta definição. Além disso, é necessário um número de expectadores, sem nenhum conhecimento de codificação de vídeo, variando entre 4 (quatro) e 40 (quarenta), sendo que é recomendado o uso de no mínimo 15. O experimento consiste na exibição do vídeo original, seguido de um intervalo de dois segundos, e então, o vídeo codificado com o algoritmo em teste é apresentado. Os observadores atribuem uma nota após a análise de ambas sequencias. As notas podem variar entre um e cinco, onde um representa a maior degradação de qualidade e cinco significa que não foi possível perceber nenhuma degradação de qualidade.

No experimento realizado, os vídeos das CCTs foram apresentados a 26 estudantes do curso de Engenharia de Computação da Universidade Federal de Pelotas. Todos os alunos estavam cursando o primeiro semestre, não possuíam problemas de visão e nenhum conhecimento na área de codificação de vídeo. As idades dos estudantes utilizados neste experimento variaram entre 17 e 22 anos.

Neste experimento, os vídeos foram apresentados em uma TV Samsung de 46 polegadas modelo UN46F6400 da série 6 (SAMSUNG, 2014) e resolução 1920x1080, ou seja, *Full* HD.

Inicialmente, os vídeos originais foram comparados com os vídeos codificados pelo HTM-7.0 tradicional (sem modificações). Após, os vídeos originais foram comparados com os vídeos codificados utilizando o DMMFast. Em ambos os experimentos, todos os vídeos das CCT foram codificados utilizando QP=30. Este QP foi escolhido por ser um QP intermediário utilizado nas CCT, não inserindo efeitos de bloco relevantes na codificação. Dessa forma, as características da geração das vistas sintetizadas podem ser melhor visualizadas, tais como a qualidade da codificação das bordas.

A Tabela 10 sumariza os resultados obtidos na análise subjetiva e, além disso, apresenta os ganhos de *bit rate* ao utilizar a solução proposta em relação aos

resultados do HTM-7.0. Para os resultados de qualidade, foram feitas as diferenças entre os resultados do HTM-7.0 e os resultados do DMMFast, sendo que resultados negativos representam melhorias de qualidade, e resultados positivos representam perdas. Para uma análise mais detalhada, a Figura 47 apresenta a média das notas de cada um dos vídeos.

Tabela 10 – Resultados da Análise Subjetiva.

| Vídeos | Média | Desvio padrão | *Bit rate* | Redução de Complexidade (tempo de codificação) |
|---|---|---|---|---|
| *Balloons* | -0,0326 | 0,4932 | -0,298% | 5,8% |
| *Kendo* | 0,0231 | 0,7637 | -0,181% | 5,7% |
| *Newspaper_CC* | -0,5192 | 0,9707 | -1,111% | 6,5% |
| *GT_Fly* | 0,0538 | 0,5329 | -0,210% | 6,2% |
| *Poznan_Hall2* | 0,8000 | 1,1839 | -0,204% | 6,3% |
| *Poznan_Street* | 0,0154 | 0,9092 | -0,610% | 7,4% |
| *Undo_Dancer* | -0,1576 | 0,6434 | 0,020% | 6,0% |
| **Média** | **0,0261** | **0,7853** | **-0,371%** | 6,3% |

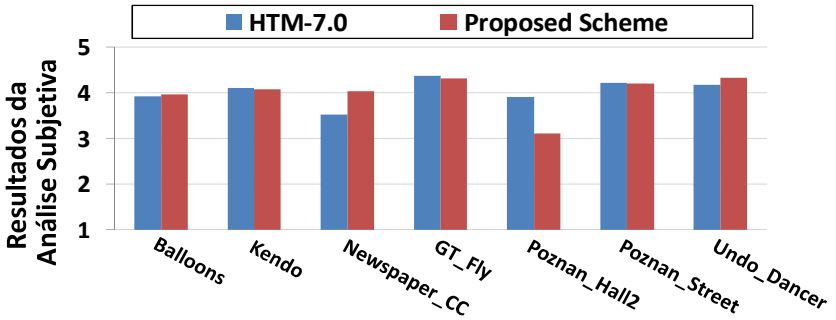



Figura 47 – Resultados da Análise Subjetiva.

Em média, o HTM-7.0 tradicional obteve 0,0261 pontos de qualidade a mais do que a solução proposta, o que representa quase nenhuma degradação de qualidade. Além disso, o DMMFast é capaz de atingir, em média, uma redução de *bit rate* de 0,371%, além de uma redução média de complexidade de 6,3%. Considerando a perda de qualidade e a redução de *bit rate* associada, é possível assumir uma diminuição do valor do QP utilizado na solução proposta, ocasionando um aumento de *bit rate*, e consequentemente, um aumento de qualidade. Com isso, pode-se dizer que a solução proposta apresenta resultados equivalentes aos do HTM-7.0. Esta análise prova que a solução proposta é capaz de obter a mesma qualidade subjetiva do HTM-7.0 original, com uma considerável redução na complexidade do codificador.

# 7 CONCLUSÕES E TRABALHOS FUTUROS

O 3D-HEVC é baseado no modelo de codificação 3D MVD, que inclui mapas de profundidade na codificação dos vídeos, e cada mapa de profundidade está associado a um quadro de textura. A codificação de profundidade é interessante, pois permite reduzir o número de vistas codificadas e fazer a síntese de vistas intermediárias no decodificador através da interpolação de vistas de textura com vistas de profundidade.

Por possuírem características distintas dos quadros de textura, novas ferramentas específicas foram criadas para a codificação de mapas de profundidade. Por um lado, estas ferramentas conseguem elevar a qualidade das vistas sintetizadas, pois conseguem preservar melhor as arestas durante o processo de codificação. Por outro lado, a codificação de profundidade insere uma complexidade adicional indesejada no codificador completo. Essas novas ferramentas são inovadoras e foram criadas recentemente, com isso, existem poucos trabalhos na literatura propondo soluções a respeito da redução de complexidade destes algoritmos.

Neste trabalho foi desenvolvido o *Depth Modeling Modes Fast Prediction* (DMMFast), um esquema de redução de complexidade focado na predição intra-quadros de mapas de profundidade do padrão emergente 3D-HEVC. O DMMFast é composto de dois algoritmos: (1) o *Simplified Edge Detector* (SED) e (2) o *Gradient-based Mode One Filter* (GMOF).

O algoritmo SED é um algoritmo capaz de classificar um bloco em aresta ou em região constante. Através desta classificação, o SED determina se os novos modos de codificação de profundidade devem ou não ser aplicados. Caso o bloco seja classificado como aresta, então o fluxo tradicional de codificação do 3D-HEVC é realizado, porém, caso o bloco seja classificado como região homogênea, os novos modos de codificação de profundidade são descartados, e apenas a predição intra tradicional do HEVC é avaliada. Este método é motivado pelo fato de que os novos modos de predição de profundidade foram desenvolvidos especialmente para melhor codificar blocos contendo arestas.

O algoritmo GMOF é responsável por reduzir drasticamente a quantidade de *Wedgelets* avaliadas pelo algoritmo do *Depth Modeling Mode* (DMM) 1, que apresenta o maior custo computacional entre todos os novos modos intra do 3D-

HEVC. Este algoritmo aplica um filtro em todas as bordas do bloco, determinando as posições mais promissoras de representarem uma aresta, e assim, apenas necessita avaliar as posições mais promissoras.

O DMMFast foi avaliado no HTM-7.0, simulado segundo as Condições Comuns de Teste (CCT) e comparado com os resultados do HTM-7.0 sem nenhuma modificação. Os resultados mostram que o DMMFast apresenta uma redução média de 6,3% na complexidade da codificação, considerando o codificador completo, ou 24,5% considerando apenas a codificação dos mapas de profundidade. Além disso, através do uso do DMMFast, é possível obter um ganho de 0,086% no BD-Rate, considerando as CCTs. Considerando o cenário All-Intra (AI), foi possível obter um ganho de complexidade médio de 64%.

Além disso, também foi realizada uma avaliação de qualidade subjetiva do DMMFast, utilizando o método *Degration Category Rating* (DCR). Nesta análise, 26 estudantes do curso de Engenharia de Computação, sem nenhum conhecimento de codificação de vídeo, avaliaram a qualidade subjetiva das vistas sintetizadas geradas com a utilização do DMMFast. Através desta análise, a solução proposta obteve uma perda de 0,0261 pontos, o que pode ser considerado uma perda insignificante. Além disso, considerando o *bit rate* da solução proposta, é possível concluir que para uma mesma qualidade de vídeo, as soluções apresentam um *bit rate* equivalente.

Através de todas as análises desenvolvidas para o DMMFast, é possível concluir que ele apresenta um ganho considerável de redução de complexidade com perdas insignificantes (ou até nulas) de eficiência de codificação.

Além do exposto neste trabalho, o 3D-HEVC proporciona uma vasta área de exploração para outros trabalhos. Considerando a grande complexidade computacional deste padrão emergente, como trabalhos futuros pretende-se fazer algumas variações no algoritmo GMOF, como por exemplo, avaliar apenas uma única reta na avaliação do DMM 1, visando reduzir ainda mais a sua complexidade. Além disso, devido ao grande esforço computacional utilizado pela predição inter-quadros, também pretendemos desenvolver soluções simplificadas, alterando o algoritmo de busca, principalmente quando a codificação de áreas homogêneas estiver sendo realizada.

O desenvolvimento de soluções em hardware focando a codificação de vídeos 3D de alta definição também é um tópico de grande interesse. Arquiteturas

de hardware dedicadas são necessárias para atingir os requisitos de desempenho para a codificação em tempo real e, até agora, nenhum trabalho relacionado a este tema pode ser encontrado na literatura. Com isso, pretendemos desenvolver arquiteturas relacionadas com os algoritmos desenvolvidos nesta dissertação e arquiteturas relacionadas com os novos modos DMMs.

## REFERÊNCIAS


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.

BHASKARAN, V.; KONSTANTINIDES, K. **Image and Video Compression Standards: Algorithms and Architectures**. 2nd ed. Boston: Kluwer Academic Publishers, 1999.

CHEN, Y.; WANG, Y. UGUR, K., HANNUKSELA, M. LAINEMA, J.; GABBOUJ, M. **The Emerging MVC Standard for 3D Video Services.** EURASIP Journal on Advances in Signal Processing. 2008, p. 1-13.

CHENG,Y. , et al. **An H.264 Spatio-Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video**. IEEE International Symposium on Circuits and Systems (ISCAS), 2009.

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, vol. 22, no. 2, dezembro de 2012.

GU, Z.; ZHENG, J.; LING, N.; ZHANG, P. **Fast Depth Modeling Mode Selection for 3D HEVC Depth Intra Coding**. International Conference on Multimedia and Expo Workshops (ICMEW), 2013a.

GU, Z.; ZHENG, J.; LING, N.; ZHANG, P. **3D-CE5.h related: Fast Intra Prediction Mode Selection for Intra Depth Map Coding.** Documento: JCT3V-E0238. Vienna, 2013b.

GUO, X.; GAO, W.; ZHAO, D. **Motion vector prediction in multiview video coding.** IEEE International Conference on Image Processing (ICIP), 2005.

HE, P.; YU, M.; PENG, Z.; JIANG, G. **Fast mode selection and disparity estimation for multiview video coding**. IEEE Int. Symp. Intell. Inf. Technol. Appl. Workshops, 2009.

ITU – International Telecommunication Union. **P.910 : Subjective video quality assessment methods for multimedia applications.** Fevereiro de 2009.

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 fevereiro de 2014.

KIM, Y.; KIM, J.; SOHN, K. **Fast Disparity Motion Estimation for Multi-View Video Coding**. IEEE Transactions on Consumer Electronics. vol. 53, no. 2, pp. 712-719, março de 2007.

KUO, T.Y.; LAI, Y.Y.; Lo, Y.C. **Fast Mode Decision for non-anchor picture in multiview video coding**. IEEE International Symposium on Broadband Multimedia System Broadcast, 2010.

LI, X.; ZHAO, D.; Ma, S.; GAO, W. **Fast Disparity Motion Estimation Based on Correlations for Multiview Video Coding**. IEEE Transactions on Consumer Electronics. vol. 54, no. 4, pp. 2037-2044, novembro de 2008.

MARPE, D.; SCHWARZ, H.; WIEGAND, T. **Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard**. IEEE Transactions on Circuits and Systems for Video Technologies, vol. 13, no. 7, pp. 620–636, julho de 2003.

MERKLE, P.; SMOLIC, A.; MULLER, K.; WIEGAND, T. **Efficient Compression of Multi-view Depth Data Based on MVC.** 3DTV Conference, 2007a.

MERKLE, P.; SMOLIC, A.; MULLER, K.; WIEGAND, T. **Multi-view Video Plus Depth Representation and Coding**. IEEE International Conference on Image Processing (ICIP), 2007b.

MERKLE, P.; et al. **3D video: Depth Coding Based on Inter-Component Prediction of Block Partitions.** IEEE Picture Coding Symposium (PCS), 2011.

MOBILE-3DTV. **Mobile-3DTV Solid Video Database**. Disponível em: <http://sp.cs.tut.fi/mobile3dtv/stereo-video/>. Acesso em 16 de julho de 2013.

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, Set. 2013.

MULLER, K.; SMOLIC, K.; DIX, P.; MERKLE, P.; KAUFF, P.; WIEGAND, T. **View Synthesis for Advanced 3D Video Systems**. EURASIP Journal on Image Video Processing, Special Issue 3D Image Video Processing, vol. 2008, no. 438148, pp. 1-11, 2008.

MULLER, K. et al. **3-D Video Representation Using Depth Maps.** Proceeding of IEEE, 2011.

MULLER, K. et al. **3D High-Efficiency Video Coding for Multi-View Video and Depth Data.** IEEE Transactions on Image Processing, 2013.

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, vol. 22, no. 12, dezembro de 2012.

OPPENHEIM, A.; WILLSKY, A. **Signal and Systems**. 2nd ed. Prentice Hall, 1996.

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

ROMBERG, J.K.; WAKIN, M.; BARANIUK, R. **Multiscale wedgelet image analysis: fast decompositions and modeling.** International Conference on Image Processing (ICIP), 2002.

RUSANOVSKYY, D.; MULLER, K.; VETRO, A. **Common Test Conditions on 3D Core experiments**. Documento: JCT3V-C1100. Geneva, 2013.

SAMSUNG. **46 F6400 Smart 3D Full HD LED TV**. Disponível em: <http://www.samsung.com/br/support/model/UN46F6400AGXZD>. Acesso em: 19 set., 2014.

SHEN, L.; LIU, Z.; ZHANG, Z.; An. P. **Selective Disparity Estimation and Variable Size Motion Estimation Based on Motion Homogeneity for Multi-View Coding**. IEEE Transactions on Broadcast. Vol 55, no. 4, pp 761-766, dezembro de 2009.

SHEN, L.; YAN, T.; ZHANG, Z.; AN, P. **Early SKIP Mode Decision for MVC Using Inter-View Correlation**. , Image Commun. Signal Processing, vol. 25, no. 2, pp. 88–93, fevereiro de 2010.

SILVA, M. **Heurísticas para Redução da Complexidade na Predição Inter-Quadros do Padrão de Codificação de Vídeo Emergente HEVC**. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação), Universidade Federal de Pelotas. Pelotas, 2013.

SONG, Y.; HO, Y. **Simplified Inter-Component Depth Modeling in 3d-Hevc**. IEEE 11th IVMSP Workshop, 2013.

SULLIVAN, G.J.; OHM, J.. HAN, W.; WIEGAND, T. **Overview of the High Efficiency Video Coding (HEVC) Standard.** IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, dezembro de 2012.

TANIMOTO, M. **Overview of FTV (FREE-VIEWPOINT TELEVISION)**. IEEE International Conference on Multimedia and Expo, New York, USA, 2009.

TECH, G.; WEGNER, K.; CHEN, Y.; YEA, S. **3D HEVC Test Model 3**. Documento: JCT3V-C1005. Draft 3 of 3D-HEVC Test Model Description. Geneva, 2013a.

TECH, G.; WEGNER, K.; CHEN, Y.; YEA, S. **3D-HEVC Draft Text 1.** Documento: JCT3V-C1005. Draft 3 of 3D-HEVC Test Model Description. Geneva, 2013b.

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, vol. 99, no. 4, abril de 2011 .

YASAKETH, S.L.P; FERNANDO, W.A.C; KAMOLRAT, B.; KONDOZ, A. **Analysing Perceptual Attributes of 3D Video**. IEEE Transactions on Consumer Electronics, vol. 55, no. 2, maio de 2009.

YANG, S.H.; LIOU, Y.S. **Fast Reference Frame and Mode Selection for Multiview Video Coding Based on Coded Block Patterns**. IEEE International Conference on Multimedia and Expo, 2010.

YEH, C.; LI, M.; CHEN, M.; CHI, M.; HUANG, X.; CHI, H. **Fast Mode Decision Algorithm Through Inter-View Rate-Distortion Prediction for Multiview Video Coding System**. IEEE Transactions on Circuits and Systems for Video Technology, 2014.

ZATT, B.; SHAFIQUE, M.; BAMPI, S.; HENKEL, J. **An Adaptive Early Skip Mode Decision Scheme for Multiview Video Coding**. Picture Coding Symposium, Nagoya, Japan, 2010.

ZHANG, Q.; LI, Z.; HUANG, L.; GAN, Y. **Effective early termination algorithm for depth map intra coding in 3D-HEVC.** IEEE Electronic Letters, vol. 50, no. 14, pp. 994-996, julho de 2014.

ZHANG. Y; Kwong, S.; Jiang, G.; Wang, X.; Yu, M. **Statistical Early Termination Model for Fast Mode Decision and Reference Frame Selection in Multiview Video Coding.** IEEE Transactions on Broadcasting, Março de 2012.

ZHAO, L.; ZHANG, L.; MA, S.; ZHAO, D. **Fast Mode Decision Algorithm for Intra Prediction in HEVC.** IEEE Visual Communications and Image Processing. Tainan, 2011.

ZHU, W.; TIAN, X.; ZHOU, F.; CHEN, Y. **Fast Disparity Estimation Using Spatio-Temporal Correlation of Disparity Field for Multiview Video Coding**. IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 957-964, maio de 2010.

3D-HEVC-HTM. **Multimedia Communications with SVC, HEVC and SHVC.** Disponível em: <http://r2d2n3po.tistory.com/66>. Acesso em 03 de maio de 2013.

## Apêndice A

**Lista das principais publicações durante o mestrado**:

Relacionadas com o 3D-HEVC:

1. Título: "**Complexity Reduction for 3D-HEVC Depth Maps Intra-Frame Prediction Using Simplified Edge Detector Algorithm**"
   Evento: International Conference on Image Processing 2014 (ICIP - Qualis A1)

2. Título: "**A Complexity Reduction Algorithm for Depth Maps Intra Prediction on the 3D-HEVC**"
   Evento: Visual Communications and Image Processing 2014 (VCIP - Qualis B1)

3. Título: "**A Real-Time 5-Views HD 1080p Architecture for 3D-HEVC Depth Modeling Mode 4**"
   Evento: Symposium on Integrated Circuits and System Design 2014 (SBCCI - Qualis B1)

4. Título: "**Overview and quality analysis in 3D-HEVC emergent video coding standard**"
   Evento: Latin American Symposium on Circuits and Systems 2014 (LASCAS – Qualis B5)

Relacionadas com a estimação de movimento do HEVC:

1. Título: "**Hardware-friendly HEVC motion estimation: new algorithms and efficient VLSI designs targeting high definition videos. Analog Integrated Circuits and Signal Processing**"
   Journal: Analog Integrated Circuits and Signal Processing 2014 (ALOG - Qualis B1)

2. Título: "**A hardware friedly motion estimation algorithm for the emergent HEVC standard and its low power hardware design**"
   Evento: International Conference on Image Processing 2013 (ICIP - Qualis A1)

3. Título: "**ES&IS: Enhanced Spread and Iterative Search hardware-friendly motion estimation algorithm for the HEVC Standard**"
   Evento: International Conference on Electronics, Circuits, and Systems 2013 (ICECS – Qualis B1)

4. Título: "**A fast hardware-friendly motion estimation algorithm and its VLSI design for real time ultra high definition applications**"
   Evento: Latin American Symposium on Circuits and Systems 2013 (LASCAS – Qualis B5)