

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



**Tese**

**Compressão de Nuvens de Pontos Dinâmicas: Uma Abordagem  
Eficiente para a Etapa de Predição**

**Cristiano Flores dos Santos**

**Pelotas, 2020**

**Cristiano Flores dos Santos**

**Compressão de Nuvens de Pontos Dinâmicas: Uma Abordagem  
Eficiente para a Etapa de Predição**

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

Orientador: Prof. Dr. Marcelo Schiavon Porto  
Coorientadores: Prof. Dr. Guilherme Ribeiro Corrêa  
Prof. Dr. Luís Alberto da Silva Cruz

Pelotas, 2020

Universidade Federal de Pelotas / Sistema de Bibliotecas  
Catalogação na Publicação

S237c Santos, Cristiano Flores dos

Compressão de nuvens de pontos dinâmicas : uma abordagem eficiente para a etapa de predição / Cristiano Flores dos Santos ; Marcelo Schiavon Porto, orientador ; Guilherme Ribeiro Corrêa, Luís Alberto da Silva Cruz, coorientadores. — Pelotas, 2020.

152 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2020.

1. Compressão. 2. Nuvens de pontos. 3. Predição. 4. Estimação de movimento 3D. I. Porto, Marcelo Schiavon, orient. II. Corrêa, Guilherme Ribeiro, coorient. III. Cruz, Luís Alberto da Silva, coorient. IV. Título.

CDD : 005

Cristiano Flores dos Santos

**Compressão de Nuvens de Pontos Dinâmicas: Uma Abordagem  
Eficiente para a Etapa de Predição**

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas.

Data da Defesa: 17 de abril de 2020

Banca Examinadora:

Prof. Dr. Marcelo Schiavon Porto (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul, UFRGS

Prof. Dr. Bruno Zatt

Doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul, UFRGS

Prof. Dr. Mateus Grellert da Silva

Doutor em Computação pela Universidade Federal do Rio Grande do Sul, UFRGS

Prof. Dr. Eduardo Antônio Barros da Silva

Doutor em Eletrônica pela Universidade de Essex, ESSEX

## RESUMO

SANTOS, Cristiano Flores dos. **Compressão de Nuvens de Pontos Dinâmicas: Uma Abordagem Eficiente para a Etapa de Predição.** 2020. 152 f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2020.

Atualmente, novas tecnologias para representação de cenas com maior realismo estão sendo alvo de intensas pesquisas. Nesse contexto, os conteúdos multimídia em 3D estão ganhando foco e possibilitando aplicações como vídeos tele-immer- sivos, realidade virtual, aumentada e mista, mapeamento de objetos históricos e monumentos arquitetônicos, entre outras. As nuvens de pontos têm atraído a atenção de pesquisas atuais como forma de representação de conteúdos 3D diante do seu baixo custo computacional considerando aspectos como geração e manipulação dos dados, destacando-se assim frente a outras alternativas, como malhas (*meshes*). Porém, assim como em vídeos digitais 2D, o volume de dados gerados por nuvem de pontos sem compressão se torna proibitivo para a sua transmissão e armazenamento. No cenário de compressão de vídeo são exploradas redundâncias, entre elas, a redundância espacial (intra), e a redundância temporal (inter). No entanto, características inerentes às nuvens de pontos, como diferentes limites das caixas delimitadoras entre nuvens temporalmente vizinhas, assim como a presença de regiões oclusas ou esparsas, tornam desafiadora a exploração dessas predições na compressão de nuvens de pontos. Nesse sentido, a compressão de nuvem de pontos dinâmica, em especial as etapas de predição intra e inter nuvens, é um tema ativo e desafiador, ainda pouco explorado na literatura. Os trabalhos encontrados na literatura que tratam da predição inter nuvens a partir de macroblocos, exploram apenas tamanho fixo e/ou exploram somente macroblocos colocalizados. Propostas mais recentes sugerem ainda planificar as nuvens de pontos e utilizar codificadores de vídeos. No entanto, com essa abordagem a exploração do espaço tridimensional acaba sendo desprezada na etapa de predição inter nuvens. Esta tese tem como objetivo apresentar uma abordagem eficiente para predição na compressão de nuvens de pontos dinâmicas. A solução proposta contempla uma estimativa de movimento baseada em macroblocos no espaço 3D, além da capacidade de operar sobre diferentes configurações de tamanhos de blocos, assim como um algoritmo de decisão de modo para escolher a melhor configuração baseada na relação entre taxa de bits e qualidade. A solução foi implementada e avaliada seguindo as condições comuns de testes propostas pelo MPEG, que contempla a avaliação de cinco sequências de nuvens de pontos dinâmicas, cada uma com aproximadamente 300 nuvens de pontos. Os resultados de taxa e qualidade foram comparados com os obtidos no software âncora para compressão de nuvem de pontos dinâmica. As abordagens desenvolvidas apresentam ganho significativo, tanto em termos de qualidade como em taxa de compressão.

**Palavras-Chave:** compressão; nuvens de pontos; predição, estimativa de movimento 3D.

## ABSTRACT

SANTOS, Cristiano Flores dos. **Compression of Dynamic Point Clouds: An Efficient Approach for the Prediction Step.** 2020. 152 f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2020.

Currently, new technologies for representing scenes with greater realism are the subject of intense research. In this context, 3D multimedia content is gaining focus and enabling applications such as tele-immersive videos, virtual, augmented and mixed reality, mapping of historical objects and architectural monuments, among other possibilities. Point clouds have attracted the attention of current research as a way of representing 3D content because of its low computational cost considering aspects such as data generation and manipulation, thus standing out against alternatives such as meshes. However, as in 2D digital videos, the volume of data generated by point clouds without compression becomes prohibitive for transmission and storage. In the video compression scenario, redundancies are explored, including spatial redundancy, called intra-frame prediction and temporal redundancy, called inter-frame. The temporal redundancy present in the videos, for example, allows the most significant gains in compression. However, characteristics inherent to the point cloud as different boundaries of the bounding boxes between temporally neighboring clouds, as well as the presence of occluded or sparse regions, makes it difficult to explore these predictions in compression. In this sense, dynamic point cloud compression, especially the intra and inter-cloud prediction stages, is an active and challenging topic, still little explored in the literature. The works found in the literature that deal with inter-cloud prediction through macroblocks, for example, explore only fixed size and/or explore only colocalized macroblocks. More recent proposals also suggest planning point clouds and using video encoders. However, with this approach, the exploration of three-dimensional space ends up being neglected in the inter-cloud prediction stage. This thesis aims to present an efficient approach to predict the compression of dynamic point clouds. The proposed solution includes a motion estimation based on macroblocks in 3D space, in addition to the ability to operate on different block size configurations, as well as a mode decision algorithm to choose the best configuration based on the relationship between bit rate and quality. The solution was implemented and evaluated following the common test conditions proposed by MPEG, which include the evaluation of five dynamic point cloud sequences, each with approximately 300 point clouds. The rate and quality results were compared with those obtained in the anchor software for dynamic point cloud compression. The approaches developed show significant gains, both in terms of quality and in compression rate.

**Keywords:** compression; point clouds; prediction, 3D motion estimation.

## LISTA DE FIGURAS

Figura 1: Exemplo de geração de malha.....	19
Figura 2: Exemplos de nuvem de pontos, Bunny (a) e Ricardo (b).....	19
Figura 3: Problemas das caixas delimitadoras (a) e voxels não ocupados (b).....	23
Figura 4: Exemplo de nuvem de pontos da sequência <i>Red and Black</i> (D'EON et al., 2016).....	29
Figura 5: Arquivo de nuvens de pontos.....	30
Figura 6: Exemplo de nuvem de pontos estática: <i>Egyptian mask</i> (a) e Arco Valentino (b).....	30
Figura 7: Exemplo de nuvem de pontos dinâmica, representada com a sequência <i>Soldier</i> (D'EON et al., 2016).....	31
Figura 8: Sistema de aquisição das nuvens de pontos (a) e nuvens de pontos geradas (b) (TULVAN; MEKURIA; LI,2016).....	33
Figura 9: Seis níveis de octree aplicado na nuvem <i>Long Dress</i> (D'EON et al., 2016). .....	36
Figura 10: Esquema de serialização da <i>octree</i> (Figura adaptada de (Kammerl, 2012)). .....	37
Figura 11: Exemplo de pesquisa do vizinho mais próximo na <i>Kd-tree</i> .....	40
Figura 12: Exemplo da abordagem ponto-a-plano .....	42
Figura 13: Visão geral do codec PCC-âncora (adaptado de (MEKURIA; BLOM; CESAR, 2017)).....	48
Figura 14: Normalização e alinhamento das caixas delimitadoras.....	49
Figura 15: Fluxo da predição inter-nuvens e decisão de modo (adaptado de (MEKURIA; BLOM; CESAR, 2017)).....	53
Figura 16: Estrutura do bitstream (MAMMOU, 2018). .....	55
Figura 17: Visão geral do processo de codificação do V-PCC (MAMMOU, 2018). ...	55
Figura 18: Visão geral do processo de decodificação do V-PCC (MAMMOU, 2018)	56

Figura 19: Nuvem de pontos projetada para os planos (ZAKHARCHENKO, 2018)..	56
Figura 20: Exemplo de imagem de projeção para geometria (a) e cor (b) (ZAKHARCHENKO, 2018) .....	57
Figura 21: Problema na ME no V-PCC (adaptado de (LI, et al., 2019)). .....	59
Figura 22: Comparativo entre V-PCC e PCC-âncora (adaptado de (GONÇALVES, et al., 2019)) .....	60
Figura 23: Visão geral da estimativa de movimento baseada em bloco para nuvem de pontos. ....	66
Figura 24: Resultados dos mapas de calor para o algoritmo FS com área de pesquisa [-7, +7] e [-15, +15].....	68
Figura 25: Exemplo de pesquisa com a adaptação do algoritmo TSS para operar no espaço tridimensional.....	72
Figura 26: Adaptações do DS para nuvens de pontos. ....	74
Figura 27: Visão geral do método proposto.....	82
Figura 28: Problema de alinhamento das caixas delimitadoras (adaptado de (Santos, et al., 2018)) .....	84
Figura 29: Ajuste das caixas delimitadoras (adaptado de (Santos, et al., 2018)).....	84
Figura 30: Fluxo da ME 3D.....	86
Figura 31: Discretização das coordenadas .....	88
Figura 32: Subparticionamento em 2 níveis. ....	89
Figura 33: Visão geral da proposta de subparticionamento de macroblocos. ....	90
Figura 34: Resultados para diferentes possibilidades de decisão de modo.....	92
Figura 35: Fluxo da decisão de modo. ....	93
Figura 36: Experimentações para definição do valor de $\lambda 1$ e $\lambda 2$ .....	97
Figura 37: Sequências de nuvens de pontos, <i>Long Dress</i> (a), <i>Loot</i> (b), <i>Red and Black</i> (c), <i>Soldier</i> (d) e <i>Queen</i> (e).....	99
Figura 38: Gráfico RD de geometria para a proposta de alinhamento .....	100
Figura 39: Gráfico RD de textura para a proposta de alinhamento .....	101

Figura 40: Gráfico RD de geometria para a proposta de subparticionamento.....	104
Figura 41: Gráfico RD de textura para a proposta de subparticionamento. ....	105
Figura 42: Gráficos RD para geometria.....	109
Figura 43: Gráficos RD para geometria.....	110
Figura 44: Gráficos RD para textura.....	112
Figura 45: Gráficos RD para textura.....	113
Figura 46: Avaliação visual da quantidade de blocos preditos com a predição inter (blocos em verde).....	118
Figura 47: Mapas de calor para macroblocos de tamanho $16 \times 16 \times 16$ voxels e algoritmo FS com deslocamento $\pm 7$ e $\pm 15$ .....	132
Figura 48: Mapas de calor para macroblocos de tamanho $32 \times 32 \times 32$ voxels e algoritmo FS com deslocamento $\pm 7$ e $\pm 15$ sem o algoritmo ICP. ....	133
Figura 49: Resultados de variantes da abordagem proposta.....	134
Figura 50: Resultados para sequência <i>Long Dress</i> de variações da proposta desenvolvida neste trabalho.....	135
Figura 51: Resultados para sequência <i>Loot</i> de variações da proposta desenvolvida neste trabalho.....	135
Figura 52: Resultados para sequência <i>Soldier</i> de variações da proposta desenvolvida neste trabalho.....	136
Figura 53: Resultados para sequência <i>Red and Black</i> de variações da proposta desenvolvida neste trabalho.....	136
Figura 54: Resultados para sequência <i>Queen</i> de variações da proposta desenvolvida neste trabalho.....	137
Figura 55: Gráficos RD de geometria para a proposta de alinhamento das caixas delimitadoras.....	138
Figura 56: Gráficos RD para textura para a proposta de alinhamento das caixas delimitadoras.....	139
Figura 57: Gráficos RD para geometria para a proposta de Subparticionamento...	140

Figura 58: Gráficos RD para textura para a proposta de Subparticionamento.....	140
Figura 59: Resultados da solução proposta para a nuvem de número 1052 da sequência <i>Long Dress</i> .....	147
Figura 60: Resultados da solução proposta para a nuvem de número 1001 da sequência <i>Loot</i> .....	148
Figura 61: Resultados da solução proposta para a nuvem de número 1451 da sequência <i>Red and Black</i> .....	148
Figura 62: Resultados da solução proposta para a nuvem de número 537 da sequência <i>Soldier</i> .....	149
Figura 63: Resultados da solução proposta para a nuvem de número 0001 da sequência <i>Queen</i> .....	149
Figura 64: Resultados das decisões de modo testadas para <i>Long Dress</i> 1052.....	150
Figura 65: Resultados das decisões de modo testadas para <i>Loot</i> 1001 .....	150
Figura 66: Resultados das decisões de modo testadas para <i>Red and Black</i> 1451.	151
Figura 67 Resultados das decisões de modo testadas para <i>Soldier</i> 537 .....	151
Figura 68: QR CODE para apresentação de resultados visuais adicionais dos passos do desenvolvimento. ....	152

## LISTA DE TABELAS

Tabela I: A estrutura dos dados que compõem o vetor de movimento. ....	51
Tabela II: Avaliação comparativa de PSNR entre o algoritmo FS com blocos de $32 \times 32 \times 32$ e $16 \times 16 \times 16$ voxels. ....	69
Tabela III: Avaliação comparativa de tempo (minutos) do algoritmo FS. ....	70
Tabela IV: Comparação de qualidade (PSNR) entre algoritmos de ME avaliados ....	76
Tabela V: Comparação de complexidade entre os algoritmos de ME avaliados. ....	77
Tabela VI: Comparativo entre o algoritmo DS com ICP e sem ICP. ....	79
Tabela VII: Percentual médio de pontos codificados por meio da predição inter-nuvens. ....	102
Tabela VIII: Resultados de <i>Bjontegaard-Delta</i> para esquema de alinhamento desenvolvido em relação ao PCC-âncora. ....	103
Tabela IX: Percentual médio de pontos codificados por meio da predição inter-nuvens para os esquemas de subparticionamento de macroblocos em relação ao PCC-âncora. ....	106
Tabela X: Resultados de <i>Bjontegaard-Delta</i> de geometria para esquema de subparticionamento desenvolvido em relação ao PCC-âncora. ....	106
Tabela XI: Resultados de <i>Bjontegaard-Delta</i> de textura para esquema de subparticionamento desenvolvido em relação ao PCC-âncora. ....	107
Tabela XII: Resultados de tempo (em minutos) para diferentes <i>LoD</i> das diferentes configurações do esquema proposto em relação ao PCC-âncora. ....	115
Tabela XIII: Resultados de <i>Bjontegaard-Delta</i> de geometria da solução completa desenvolvida em relação ao PCC-âncora. ....	115
Tabela XIV: Resultados de <i>Bjontegaard-Delta</i> em textura para a solução completa desenvolvida em relação ao PCC-âncora. ....	116
Tabela XV: Número médio de pontos preditos como inter, separado por <i>LoD</i> . ....	117
Tabela XVI: Avaliação dos algoritmos de ME para macroblocos de tamanho 32....	129

Tabela XVII: Avaliação dos algoritmos de ME para macroblocos de tamanho 16 ...	129
Tabela XVIII: Avaliação dos algoritmos de ME SEM ICP para macroblocos de tamanho 32. ....	130
Tabela XIX: Avaliação dos algoritmos de ME SEM ICP para macroblocos de tamanho 16. ....	130
Tabela XX: Avaliação comparativa de tempo (minutos) dos algoritmos entre FS e DS com ICP e tamanho de macroblocos 32x32x32 <i>voxels</i> em comparação aos respectivos algoritmos sem o ICP.....	130
Tabela XXI: Avaliação comparativa de tempo (minutos) dos algoritmos entre FS e DS com ICP e tamanho de macroblocos 16x16x16 <i>voxels</i> em comparação aos respectivos algoritmos sem o ICP.....	131
Tabela XXII: Avaliação comparativa de qualidade de geometria dos algoritmos entre FS e DS com ICP e também sem ICP para macroblocos 16x16x16.....	131
Tabela XXIII: Resultados de <i>Bjontegaard-Delta</i> da abordagem desenvolvida, e o trabalho anterior em relação ao <i>PCC-âncora</i> .....	137
Tabela XXIV: Resultados de <i>Bjontegaard-Delta</i> da abordagem desenvolvida, e o trabalho anterior em relação ao <i>PCC-âncora</i> .....	138
Tabela XXV: Resultados para os dados de geometria do alinhamento das caixas delimitadoras em comparação ao <i>PCC-âncora</i> .....	141
Tabela XXVI: Resultados para os dados de textura do alinhamento das caixas delimitadoras em comparação ao <i>PCC-âncora</i> .....	142
Tabela XXVII: Resultados para os dados de geometria do subparticionamento de macroblocos em comparação ao <i>PCC-âncora</i> .....	143
Tabela XXVIII: Resultados para os dados de textura do subparticionamento de macroblocos em comparação ao <i>PCC-âncora</i> .....	144
Tabela XXIX: Resultados comparativo de qualidade de geometria e taxa de bits para a solução proposta iniciando com macroblocos 16x16x16, a solução iniciando com macroblocos 32x32x32 e o <i>PCC-âncora</i> .....	145

Tabela XXX: Resultados comparativo de qualidade de textura e taxa de bits para a solução proposta iniciando com macroblocos 16x16x16, a solução iniciando com macroblocos 32x32x32 e o PCC-âncora..... 146

## LISTA DE ABREVIATURAS E SIGLAS

2D	Duas Dimensões
3D	Três Dimensões
3SS	<i>Three Step Search</i>
4SS	<i>Four Step Search</i>
AVC	<i>Advanced Video Coding</i>
BDM	<i>Block Distortion Metric</i>
BD	<i>Bjontegaard-Delta</i>
bpp	<i>bits per point</i>
Cb	<i>Chrominance Blue</i>
Cr	<i>Chrominance Red</i>
CTC	<i>Common Test Conditions</i>
DS	<i>Diamond Search</i>
FS	<i>Full Search</i>
FTV	<i>Free-viewpoint TV</i>
G-PCC	<i>Geometry-based Point Cloud Compression</i>
GOC	<i>Group of Clouds</i>
HEVC	<i>High Efficiency Video Coding</i>
ICP	<i>Iterative Closest Point</i>
ITU-T	<i>International Telecommunication Union – Telecommunication</i>
Kd-tree	<i>k-dimensional tree</i>
K-nn	<i>k-nearest neighbors</i>
LDSP	<i>Large Diamond Search Pattern</i>
LIDAR	<i>Light Detection and Ranging</i>
LoD	<i>Level of Detail</i>
Luma	Luminância
MB	Macrobloco
MBS-8	Macrobloco Subparticionado de tamanho 8
MBS-16	Macrobloco Subparticionado de tamanho 16
MPEG	<i>Moving Picture Experts Group</i>
MSE	<i>Mean Squared Error</i>
ME	<i>Motion Estimation</i>
MC	<i>Motion Compensation</i>
MV	<i>Motion Vector</i>

NSS	<i>N Step Search</i>
VM	<i>Vetor de Movimento</i>
P2P	<i>Point-to-point</i>
P2Plane	<i>Point-to-plane</i>
PCC	<i>Point Cloud Compression</i>
PCL	<i>Point Cloud Library</i>
PLY	<i>Polygon File Format</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
RAHT	<i>Region-Adaptive Hierarchical Transform</i>
RD	<i>Rate and Distortion</i>
RGB	<i>Red, Green, Blue</i>
RGB-D	<i>Red, Green, Blue and Depth</i>
SAR	<i>Synthetic Aperture Radar</i>
SD	<i>Square Distance</i>
SDSP	<i>Small Diamond Search Pattern</i>
SED	<i>Squared Euclidean Distance</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
TMC	<i>Test Model Category</i>
TSS	<i>Three Step Search</i>
V-PCC	<i>Video-based Point Cloud Compression</i>
Y	<i>Luminance</i>
YCbCr	<i>Luminance, Chrominance Blue, Chrominance Red</i>

# SUMÁRIO

1	INTRODUÇÃO .....	18
1.1	Aplicações .....	21
1.2	Desafios .....	22
1.3	Principais Hipóteses .....	24
1.4	Objetivos .....	25
1.5	Principais Contribuições .....	25
1.6	Estrutura do texto .....	26
2	FUNDAMENTOS DE COMPRESSÃO DE NUVENS DE PONTOS .....	28
2.1	Conceitos Básicos de Nuvens de Pontos.....	28
2.1.1	Nuvens de Pontos Estáticas.....	30
2.1.2	Nuvens de Pontos Dinâmicas.....	31
2.1.3	Nuvens de Pontos de Aquisição Dinâmica .....	32
2.2	Redundância de Informação em Nuvens de Pontos .....	33
2.3	Compressão com Perdas e sem Perdas .....	34
2.4	<i>Octree</i> .....	35
2.5	Avaliação de Qualidade de Nuvens de Pontos .....	37
2.5.1.	Qualidade de Geometria .....	38
2.5.2.	Qualidade de Cor .....	44
3	CODIFICADORES MPEG .....	47
3.1	<i>Point Cloud Compression</i> (PCC-âncora).....	47
3.1.1	Predição Inter-nuvens .....	48
3.1.2	Predição Intra-nuvens .....	53
3.2	<i>Video-Based Point Cloud Compression</i> (V-PCC) .....	54
3.3	Considerações sobre os codificadores do MPEG .....	58
4	PREDIÇÃO INTER-NUVENS .....	61
4.1	Estado da Arte na Predição Inter-nuvens .....	62
4.2	Avaliação da Correlação Temporal em Nuvens de Pontos.....	65
5	AVALIAÇÃO DE ALGORITMOS DE ESTIMAÇÃO DE MOVIMENTO 3D EM NUVENS DE PONTOS .....	71
5.1	Algoritmo <i>N-Step Search</i> (NSS) .....	72
5.2	Algoritmo <i>Diamond Search</i> (DS) .....	73

5.3	Avaliação Comparativa.....	74
5.4	Avaliação do impacto do ICP.....	78
6	ESQUEMA DE PREDIÇÃO DESENVOLVIDO .....	80
6.1	Alinhamento das caixas delimitadoras .....	83
6.2	Estimação de Movimento 3D.....	85
6.3	Discretização das coordenadas do macroblocos .....	87
6.4	Método de Subparticionamento de MacroBlocos.....	88
6.5	Decisão de Modo.....	91
7	RESULTADOS EXPERIMENTAIS .....	98
7.1	Base Experimental Utilizada.....	98
7.2	Resultados do Ajuste das Caixas Delimitadoras .....	100
7.3	Resultados do Subparticionamento de MacroBlocos.....	103
7.4	Resultados da Solução Completa .....	108
8	CONCLUSÕES .....	119
8.1	Trabalhos Futuros.....	120

# 1 INTRODUÇÃO

Nos últimos anos houve um aumento na demanda por vídeos e imagens em três dimensões (3D). Isso ocorreu devido ao surgimento de novas tecnologias que possibilitaram a captura e visualização desse tipo de mídia, tais como os sensores do tipo RGB-D (canais de textura de cor vermelha (R), verde (G) e azul (B) com informação de profundidade (D)), óculos de realidade virtual ou mista, entre outros dispositivos.

Os vídeos e as imagens 3D proporcionam aos usuários maior realismo e interação com cenários ou objetos capturados. Além disso, podem beneficiar diversas outras áreas do conhecimento como, por exemplo, a arquitetura, por meio da captura (mapeamento) de prédios, casas e centros urbanos para conceber projetos arquitetônicos, a preservação da história, a partir do registro e estudo de artefatos e monumentos históricos e culturais, ou ainda a área da robótica, para prover a autonomia de dispositivos robóticos ou veículos.

Os conteúdos 3D se popularizaram principalmente por meio dos filmes de animações e jogos eletrônicos, onde a forma mais utilizada para representação são as malhas (*meshes*). Uma malha é composta por um conjunto de vértices, arestas e faces, que formam um polígono, que por sua vez constituem superfícies, e assim formam o objeto 3D conforme ilustrado na Figura 1. Porém, o procedimento de geração de uma malha demanda muito esforço computacional para realizar cálculos para unir os pontos. Além disso, é necessário armazenar informações tanto dos pontos, quanto informações adicionais para geração das superfícies. Assim, a malha é mais adequada para aplicações de computação gráfica.

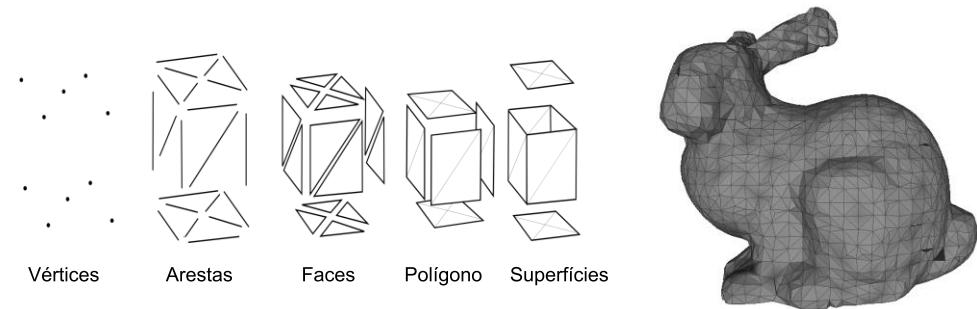


Figura 1: Exemplo de geração de malha.

As nuvens de pontos surgiram como alternativa mais viável especialmente por possibilitar a captura ou mapeamento 3D de cenários em tempo real (QUEIROZ; CHOU, 2016). Além disso, é uma forma de representação 3D simplificada que oferece realismo e qualidade com baixo custo computacional em comparação às malhas.

Uma nuvem de pontos é um conjunto de pontos compostos por informação geométrica em um espaço 3D, ou seja, coordenadas X, Y e Z, e podem conter ainda atributos de cor. Uma nuvem de pontos pode ser classificada como voxelizada quando possuir uma distribuição regular dos pontos constituindo uma grade 3D (D'EON et al., 2016). Nesse contexto, cada elemento da grade é chamado de *voxel* (unidades similares aos pixels de uma imagem).

A Figura 2, ilustra dois exemplos de nuvens de pontos. A nuvem de pontos denominada Bunny (Figura 2 (a)) possui uma representação esparsa dos pontos e sem informação de cor (textura). A nuvem de pontos Ricardo (Figura 2(b)) apresenta uma estrutura de pontos densa com atributos de cor.

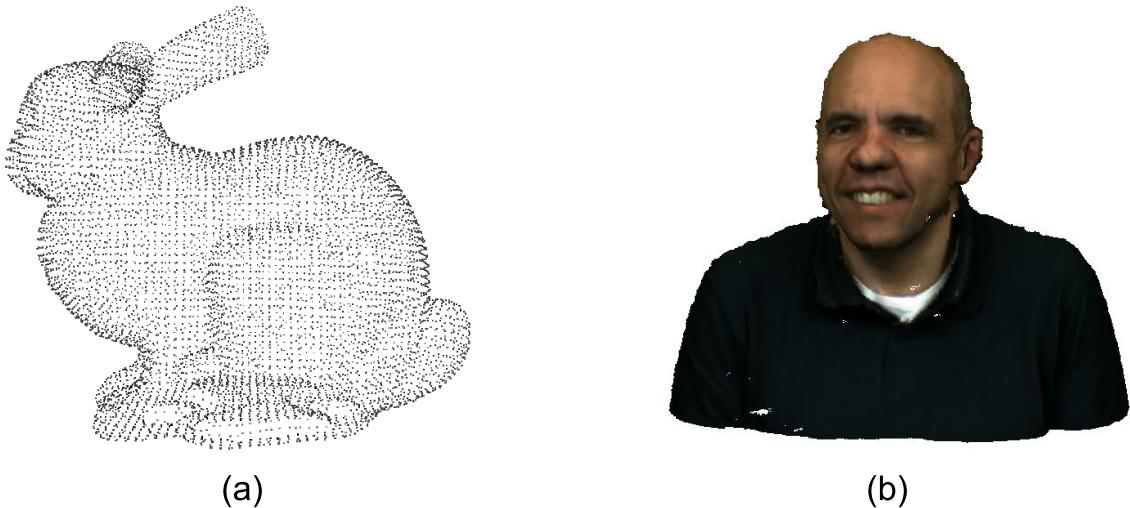


Figura 2: Exemplos de nuvem de pontos, Bunny (a) e Ricardo (b).

A geração de conteúdos 3D por meio de nuvem de pontos tornou-se mais atrativa devido à popularização e baixo custo de dispositivos de captura do tipo RGB-D, como, por exemplo, do sensor *kinect* da Microsoft (MORELL, et al., 2014), que pode ser utilizado para realizar mapeamento 3D de objetos, pessoas ou mesmo cenários (MORENO; CHEN; LI, 2017). Aliado a isso, a nuvem de pontos é mais vantajosa frente à malha, por possuir um processo de geração de conteúdos 3D mais simplificado (TULVAN; MEKURIA; LI, 2016).

A nuvem de pontos pode ser caracterizada como estática, dinâmica ou de captura dinâmica. Nuvens estáticas representam um instante de tempo, assim como uma foto. As nuvens dinâmicas são compostas por um conjunto de nuvens estáticas capturadas em diferentes instantes de tempo, de forma análoga aos vídeos digitais (2D). Já as nuvens de captura dinâmica são geradas durante a navegação e mapeamento de ambientes (TULVAN; MEKURIA; LI, 2016).

Uma nuvem de pontos pode conter milhares ou até bilhões de pontos, dependendo do nível de detalhamento ou tamanho do cenário que se pretende representar (TULVAN; MEKURIA; LI, 2016). Porém, apesar de possibilitar a representação com melhor qualidade e realismo, o número elevado de pontos é um aspecto negativo devido ao alto custo de armazenamento e transmissão. Uma nuvem de pontos dinâmica, como, por exemplo, as nuvens de pontos sugeridas nas condições comuns de teste (*Common Test Conditions - CTC*) para compressão conforme o (*Moving Picture Experts Group – MPEG*), tem em média 800 mil pontos por nuvem (quadro) (SCHWARZ; CHOU; SINHAROY, 2018). Cada ponto demanda 10 bits por componente de geometria (X, Y e Z) e 8 bits por canal de cor RGB. Assim, para armazenar 10 minutos de vídeo são necessários aproximadamente 90 Gigabytes (GBs). Para transmitir essa mesma nuvem a uma taxa de 30 nuvens por segundo (nps), seriam necessários 1.3 Gigabits por segundo (Gbps).

Nesse contexto, diante do volume de dados gerados por uma nuvem de pontos dinâmica, a compressão se torna fundamental principalmente para aplicações que necessitam transmissão. Os principais codificadores de vídeo exploram diferentes tipos de redundâncias para reduzir o volume de dados. Embora existam propostas de compressores para nuvens de pontos dinâmicas, as taxas de compressão ainda podem alcançar melhores resultados.

## 1.1 Aplicações

As nuvens de pontos são promissoras para diversas aplicações 3D que necessitam transmissão em tempo real, tais como, videoconferências e *holoportation* (aplicação que possibilita rever momentos registrados no passado de forma mais realista e interativa). Dentre as possibilidades de uso das nuvens de pontos conforme (TULVAN; MEKURIA; LI, 2016), podem-se destacar:

- **Vídeos 3D tele-imersivos em tempo real:** Permitem uma convergência entre realidade e realidade virtual, onde as nuvens de pontos são capturadas, comprimidas, transmitidas e renderizadas em tempo real similarmente a uma videoconferência.
- **Realidade virtual:** Fornece aos usuários, equipados com óculos de realidade virtual, uma experiência imersiva com um amplo campo de visão para conteúdos específicos, como filmes ou jogos digitais.
- **Free Viewpoint Television:** É a captura de nuvens de pontos de eventos esportivos, como basquete e beisebol, ou ainda de filmes para reprodução e interação tanto em dispositivos móveis quanto TVs. Isso requer um padrão de codificação e transmissão de maneira interoperável.
- **Sistemas de informação geográfica:** Por meio das nuvens de pontos é possível representar informações geográficas capturadas de sensores como *Light Detection And Ranging (Lidar)* e *Synthetic Aperture Radar (SAR)*. Esses dados geralmente são armazenados em servidores que fornecem um serviço de renderização remota ou consultas baseadas em informações geográficas específicas.
- **Patrimônio cultural:** As nuvens de pontos podem ser usadas para viabilizar o armazenamento e a visualização de coleções de objetos de patrimônio cultural. A compactação e a transmissão de nuvens de pontos representando esses objetos podem torná-los disponíveis para o público em geral.
- **Navegação autônoma baseada em mapas 3D de grande escala:** Possibilita navegação a partir de mapas 3D gerados por dispositivos que fornecem informação de profundidade juntamente com sensores de localização. Sistemas de mapeamento já estão comercialmente disponíveis

em várias formas, a exemplo de drones ou ainda smartphones/tablets que realizam o mapeamento 3D.

Considerando as possibilidades de aplicações citadas, pode-se perceber que este novo tipo de mídia é relevante para diversas áreas e contextos. Todas essas aplicações práticas de nuvens de pontos dependem da compressão eficiente dessas informações. No entanto, as nuvens de pontos apresentam características que trazem inúmeros desafios para a sua compressão de forma eficiente. Com isso, existe um vasto campo de pesquisa a ser explorado, onde os principais desafios (apresentados na próxima seção) devem ser atacados e resolvidos.

## 1.2 Desafios

Em nuvens de pontos dinâmicas, assim como em vídeos 2D, sucessivos quadros compartilham similaridades. Neste sentido, a estimativa de movimento presente na predição inter-nuvens é essencial para a compressão efetiva das nuvens de pontos dinâmicas (THANOU; CHOU; FROSSARD, 2016) (Li; Li et al.; 2019).

A Estimação de Movimento (*Motion Estimation* - ME) tem como objetivo encontrar a melhor correspondência entre blocos de nuvens temporalmente vizinhas por meio de uma métrica de similaridade. Para cada bloco da nuvem corrente são avaliados blocos candidatos em uma área de busca e, para o bloco de melhor correspondência na nuvem de referência, é calculado o vetor de movimento. Na reconstrução da nuvem, esse vetor tem como objetivo compensar o movimento do bloco predito. Em codificação de nuvens de pontos dinâmicas existe a necessidade de calcular o vetor de movimento antes de calcular a similaridade dos blocos candidatos. Isso porque as coordenadas dos pontos dos blocos avaliados devem assumir as mesmas posições do bloco atual. Somente dessa forma é possível avaliar a qualidade da informação de geometria desses blocos, já que as métricas de avaliação de nuvens de pontos consideram a distância entre os pontos. Nesse sentido, o principal desafio é calcular o vetor de movimento de forma eficiente, pois esse é responsável por determinar a melhor correspondência e também deve ser utilizado na etapa de reconstrução da nuvem.

Para nuvens de pontos dinâmicas, a predição inter-nuvens é um problema ainda em aberto, principalmente em consequência das características dos dados (Li; Li et al.; 2019). Diferentemente dos vídeos 2D, que podem ser representados por uma

matriz de pixels de tamanho fixo, as nuvens de pontos possuem uma estrutura irregular como pode ser visto na Figura 3(a), onde são ilustradas duas nuvens de pontos temporalmente vizinhas sobrepostas. Cada uma dessas nuvens de pontos possui uma caixa delimitadora (*bounding box*) que é constituída por arestas que respeitam os limites mínimos e máximos da nuvem nos seus eixos (X, Y e Z) (MEKURIA; BLOM; CESAR, 2017). Através das caixas delimitadoras ilustradas na Figura 3(a), é possível notar que nuvens temporalmente vizinhas podem não possuir as mesmas dimensões.

Outra característica distinta das nuvens de pontos é a possibilidade de existirem regiões descontínuas, ou seja, *voxels* não ocupados, como pode ser visto no exemplo da Figura 3(b). A região destacada na Figura 3(b) não possui informações de geometria (coordenadas X, Y e Z), nem mesmo atributos de cor.

As características das nuvens de pontos destacadas podem gerar problemas para a predição inter em nuvens de pontos, em especial para a ME, já que é necessário considerar a variação do número de pontos entre blocos de nuvens temporalmente vizinhas.

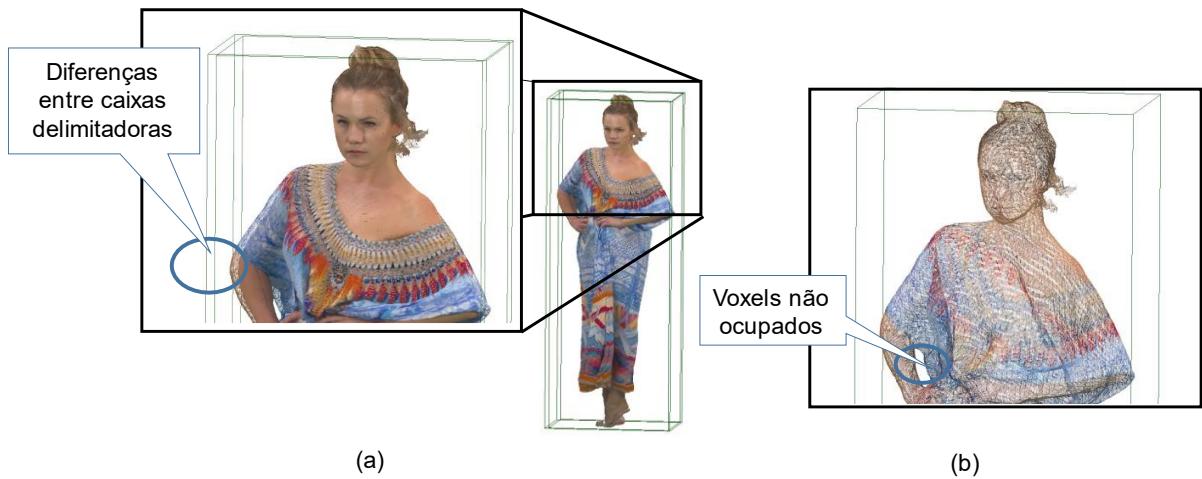


Figura 3: Problemas das caixas delimitadoras (a) e *voxels* não ocupados (b).

Outro aspecto importante, tratado na ME em codificadores de vídeo como H.264/AVC (*Advanced Video Coding*) e HEVC (*High Efficient Video Coding*), é a possibilidade de usar blocos de tamanho variável (RAO; HWANG; KIM, 2017) (SULLIVAN et al., 2012) (SULLIVAN; WIEGAND, 2005). A escolha do melhor bloco é realizada a partir de alguma métrica que conduza a uma codificação mais eficiente, de acordo com o conteúdo da região do quadro a ser codificado. O uso de diferentes tamanhos de bloco possibilita uma predição mais eficiente. Nesse contexto, a predição

de regiões homogêneas pode ser mais eficiente a partir do uso de blocos grandes, e a utilização de blocos menores tende a ser mais eficiente para regiões compostas por um maior nível de textura ou bordas.

No cenário atual, o *codec* denominado V-PCC (*Video-based Point Cloud Compression*) tem sido utilizado pelo MPEG como base para experimentação, o qual tem apresentado excelentes resultados para codificação (MAMMOU, 2018). Outro *codec* de destaque é o denominado PCC-âncora (*Point Cloud Compression*), proposto por (MEKURIA; BLOM; CESAR, 2017). O *PCC-âncora* foi utilizado pelo MPEG como âncora para comparações de novas estratégias de compressão de nuvens de pontos dinâmicas. Ambos codificadores não exploram os aspectos levantados, como blocos de tamanho variável ou predição inter-quadros considerando os aspectos 3D presentes nas nuvens de pontos. A predição inter-nuvens explorada no PCC-âncora é limitada a blocos colocalizados e dependente do alinhamento perfeito das caixas delimitadoras. O V-PCC realiza a mesma predição inter-quadros aplicada na codificação de vídeos 2D do padrão HEVC, sendo que ela é aplicada às nuvens de pontos depois de uma etapa de planificação, onde a nuvem de pontos 3D é transformada em uma imagem em 2D. O MPEG inclusive realizou uma chamada por novas estratégias de ME que contemplem as características dos dados 3D das nuvens de pontos (ZAKHARCHENKO; KIM, 2018). No entanto, a solução atual que visa tratar desse aspecto ainda apresenta limitações como ME 3D em blocos colocalizados (LI, et al., 2019).

### 1.3 Principais Hipóteses

Dado o que foi exposto como problemas ou desafios para compressão de nuvens de pontos dinâmicas, como hipótese que norteia esta tese está a ideia de que a exploração eficiente da predição pode aumentar a eficiência do processo de compressão. Em especial, a predição inter-nuvens pode ser muito beneficiada ao explorar a ME no espaço 3D. Outra hipótese é a exploração do particionamento de blocos durante o processo de predição, tanto intra quanto inter-nuvens. Esta abordagem deve prover redução de taxa em regiões homogêneas, onde blocos maiores podem ser utilizados na codificação, bem como possibilitar o refinamento da predição, tanto intra como inter, em regiões com maior variação de texturas. Para que

estas opções de codificação sejam utilizadas de maneira eficiente pelo codificador, uma decisão de modo também será necessária, para garantir a escolha da opção mais eficiente dentre as possibilidades de codificação, levando em conta os impactos tanto nas informações de geometria quanto de textura.

## 1.4 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de um novo esquema de predição para a codificação de nuvem de pontos dinâmica, visando o aumento da eficiência de codificação do codificador PCC-âncora.

Como objetivos específicos estão:

- Desenvolver e avaliar diferentes possibilidades de algoritmos rápidos para a ME;
- Propor uma nova ME baseada no espaço tridimensional para melhor explorar a correlação temporal;
- Desenvolver uma abordagem de particionamento de blocos para viabilizar a exploração de correlação;
- Propor uma decisão de modo que viabilize a escolha da melhor opção entre as possibilidades de modo de predição e tamanho de blocos;
- Apresentar os resultados tendo em vista as CTC propostas pelo MPEG.

## 1.5 Principais Contribuições

Dentre as principais contribuições deste trabalho estão:

- **Alinhamento das caixas delimitadoras:** Para explorar a correlação temporal, foi proposto um alinhamento das caixas delimitadoras para possibilitar uma correspondência mais eficiente para blocos colocalizados, bem como para viabilizar que blocos presentes nas bordas possam ter blocos candidatos colocalizados e sejam melhor avaliados.
- **Avaliação da correlação temporal em nuvens de pontos dinâmicas:** Foi proposta uma avaliação da correlação por meio de um algoritmo de busca completa para obter um melhor entendimento das correlações temporais em nuvens de pontos dinâmicas.

- **Uso de ME em 3D:** A ME 3D para nuvens de pontos possibilita ganhos em eficiência de compressão. No entanto, a ME possui um alto custo computacional. Nesse sentido, algoritmos rápidos de ME, (DS -*Diamond Search*) e (NSS - *N-Step Search*), clássicos na compressão de vídeos 2D, foram adaptados e avaliados para o contexto de nuvem de pontos 3D.
- **Predição com múltiplos tamanhos de bloco:** Foi desenvolvida uma abordagem de subparticionamento de blocos para proporcionar juntamente com a ME ganhos em compressão principalmente para grandes regiões da nuvem que não possuam ou ainda que possuam pouca alteração entre nuvens temporalmente vizinhas. O subparticionamento proposto também beneficiou a qualidade em regiões de blocos menores que possuem grande variação espacial e temporal. Além disso, essa abordagem pode ser uma alternativa para o problema de variação do número de pontos entre blocos de nuvens temporalmente vizinhas, já que os blocos com grande diferença puderam ser particionados e, assim, serem bons candidatos para a predição inter.
- **Decisão de modo com ponderação entre geometria e textura:** Foi proposto um algoritmo de decisão de modo que possibilita viabilizar a melhor escolha dentre as opções macrobloco ou bloco subparticionado, assim como o modo de codificação (intra ou inter). A decisão de modo pode avaliar aspectos como qualidade da geometria, textura e taxa, para viabilizar maior eficiência e garantir a melhor relação entre taxa e qualidade.
- **Análise comparativa dos resultados de eficiência de compressão em relação ao PCC:** Para avaliação comparativa, cinco sequências de nuvem de pontos dinâmicas sugeridas pelo MPEG foram utilizadas para avaliar a solução proposta. Os resultados que medem a qualidade e taxa de codificação foram calculados de acordo com as CTC propostas pelo MPEG e foram utilizados para comparar com o software PCC-âncora.

## 1.6 Estrutura do texto

O capítulo 2 desta tese apresenta fundamentos básicos de compressão de nuvens de pontos. Estes conceitos são importantes para o entendimento do restante

do texto. O capítulo 3 apresenta as principais soluções de compressão de nuvens de pontos dinâmicas adotadas pelo MPEG. No Capítulo 4 são apresentados alguns conceitos sobre predição inter-nuvens, em especial a estimação de movimento. Nesse capítulo é apresentada também uma avaliação da correlação temporal para as nuvens de pontos dinâmicas para uma compreensão sobre as peculiaridades do espaço 3D. O capítulo 5 apresenta uma avaliação dos algoritmos de estimação de movimento, o DS (*Diamond Search*) e NSS (*N-Step Search*), clássicos em compressão de vídeos, desenvolvidos e adaptados neste trabalho para nuvens de pontos dinâmicas. O capítulo 6 apresenta a descrição das soluções algorítmicas desenvolvidas. No capítulo 7 são apresentados os resultados das soluções desenvolvidas nesta tese. Por fim, o capítulo 8 apresenta as conclusões do trabalho destacando algumas frentes de pesquisas para trabalhos futuros.

## 2 FUNDAMENTOS DE COMPRESSÃO DE NUVENS DE PONTOS

Neste capítulo serão apresentados os principais conceitos sobre nuvens de pontos, bem como os aspectos fundamentais da sua codificação. Além disso, serão apresentados os principais trabalhos encontrados na literatura para a codificação de nuvens de pontos. Este capítulo primeiramente apresentará as características das nuvens de pontos. Na sequência, serão tratados alguns conceitos de compressão de dados digitais de forma mais abrangente e, posteriormente, no âmbito da compressão de nuvens de pontos. Também serão apresentados alguns conceitos em relação à avaliação da qualidade de nuvens de pontos codificadas. Por fim, serão apresentados os principais trabalhos relacionados à compressão de nuvens de pontos.

### 2.1 Conceitos Básicos de Nuvens de Pontos

Uma nuvem de pontos é composta por um conjunto de pontos no espaço 3D que constituem uma representação tridimensional da superfície de um objeto ou cenário. Os pontos possuem informações geométricas que são as coordenadas dos pontos num sistema ortogonal X, Y e Z e podem conter também atributos como cor, transparência e tipo de material. As cores podem ser representadas nos espaços de cor RGB ou YCbCr (luminância, crominância azul e crominância vermelha). Na Figura 4 é ilustrada uma nuvem de pontos densa e um zoom de aproximação para visualização dos pontos.

As nuvens de pontos podem ser geradas a partir de sensores do tipo RGB-D (sensores que capturam informação de textura mais profundidade) como o *Kinect* da *Microsoft* (LOOP et al., 2016), por meio de sensores LIDAR (que capturam somente a distância entre o sensor e os pontos de reflexão do feixe na superfície dos objetos) (BRIDGET, 2016) ou ainda por meio de múltiplas câmeras e sensores (D'EON et al., 2016).



Figura 4: Exemplo de nuvem de pontos da sequência *Red and Black* (D'EON et al., 2016).

As nuvens de pontos ou as sequências de nuvens de pontos que contemplam as CTC, são armazenadas no formato conhecido como (*Polygon File Format – PLY*). Esses arquivos possuem um cabeçalho com informações auxiliares e informações dos pontos, onde cada linha compreende um ponto com as informações das coordenadas dos eixos X, Y e Z seguidas dos atributos de cor separado em três canais (*RGB*), conforme ilustra a Figura 5.

A Figura 5 ilustra um exemplo de arquivo de uma nuvem de pontos no formato *PLY*, onde segue em destaque a parte do arquivo referente ao cabeçalho e a parte do arquivo com as coordenadas dos pontos com seus respectivos atributos.

As nuvens de pontos, conforme sugere o MPEG, são classificadas de acordo com as características de captura e de aplicação pretendida, sendo divididas em três categorias: nuvens estáticas, dinâmicas e de aquisição dinâmica (TULVAN; MEKURIA; LI, 2016). É importante destacar que uma nuvem de pontos do tipo dinâmica, que será apresentada em detalhes na Seção 2.1.1, contempla uma sequência de quadros análoga ao vídeo. Essa classificação de nuvem de pontos é composta por arquivos de quadros individuais no formato *PLY*.

```

cabeçalho
do arquivo
1  ply
2  format ascii 1.0
3  comment Version 2, Copyright 2017, 8i Labs, Inc.
4  comment frame_to_world_scale 0.179523
5  comment frame_to_world_translation -45.2095 7.18301 -54.3561
6  comment width 1023
7  element vertex 795809
8  property float x
9  property float y
10 property float z
11 property uchar red
12 property uchar green
13 property uchar blue
14 end_header
15 246 30 95 112 97 83
16 246 31 95 112 98 83
17 247 30 95 113 97 83
18 247 31 94 114 99 83
19 247 31 95 113 98 83

```

Figura 5: Arquivo de nuvens de pontos.

### 2.1.1 Nuvens de Pontos Estáticas

Uma nuvem de pontos estática, assim como uma imagem 2D, é uma captura de um dado objeto ou cenário em um instante de tempo, no entanto, com informações em 3D. Como exemplo de aplicação de nuvem de pontos estática é possível destacar o mapeamento 3D de patrimônios culturais históricos, conforme ilustra a Figura 6.

Os *datasets* disponíveis desse tipo de nuvens de pontos, em geral, representam somente um objeto com um número de pontos relativamente grande (centenas de milhares ou até dezenas de milhões). Muitos desses objetos são nuvens de pontos adquiridas para os mapeamentos de monumentos históricos, estátuas, ou também prédios.



(a)



(b)

Figura 6: Exemplo de nuvem de pontos estática: *Egyptian mask* (a) e Arco Valentino (b).

As nuvens de pontos estáticas são classificadas como Categoria 1 conforme o MPEG. Para a categoria 1 existe uma proposta de codificador voltado a explorar

soluções de compressão juntamente com a Categoria 3, que será apresentada na Seção 2.1.3. Esse codificador é denominado *Test Model Category 13* (TMC13). O TMC13 nas últimas iterações do MPEG passou a ser chamado (*Geometry-based Point Cloud Compression - G-PCC*) (CAO; PREDA; ZAHARIA, 2019).

### 2.1.2 Nuvens de Pontos Dinâmicas

Nuvens de pontos dinâmicas são sequências temporais de nuvens de pontos estáticas. As nuvens de pontos dinâmicas podem ser adquiridas a partir de sensores do tipo RGB-D que habilitam a aquisição simultânea de geometria e cor (MORENO; CHEN; LI, 2017).

As nuvens de pontos podem ser adquiridas também com a combinação de sensores RGB e RGB-D, assim como encontrado no *dataset* proposto por (D'EON et al., 2016). Um exemplo de nuvem de pontos dinâmicas pode ser visto na Figura 7 em que cada imagem representa uma nuvem de pontos e as setas indicam a ordem temporal.



Figura 7: Exemplo de nuvem de pontos dinâmica, representada com a sequência *Soldier* (D'EON et al., 2016).

As nuvens de pontos dinâmicas se destacam devido às possibilidades de aplicações futuras, em especial a utilização em videoconferência, proporcionando uma melhor interação entre os usuários. Outra aplicação é denominada *Holoportation*, que empresas como a *Microsoft* tem investido esforços para viabilizar não só a experiência de aproximar usuário fisicamente distantes, mas também com a possibilidade de proporcionar aos utilizadores rever cenas, como reuniões, ou ainda, reviver momentos em família de forma mais realística. Essas aplicações, podem ser

melhor exploradas por meio de tecnologias como óculos de realidade mista, ou seja, realidade aumentada juntamente com realidade virtual.

As nuvens de pontos do tipo dinâmica conforme o MPEG, pertencem à Categoria 2. Essa categoria possui como proposta de codificador, o software denominado *Video-based Point Cloud Compression* (V-PCC), também referenciado em alguns documentos do MPEG como *Test Model Category 2* (TMC2) (SCHWARZ et al, 2018). Além do V-PCC, o software denominado *Point Cloud Compression* (PCC) foi adotado previamente pelo MPEG como âncora para o desenvolvimento e experimentação de abordagens para compressão de nuvens de pontos dinâmicas (MEKURIA; CESAR, 2016) (MEKURIA; BLOM; CESAR, 2017).

### 2.1.3 Nuvens de Pontos de Aquisição Dinâmica

As nuvens de pontos de aquisição dinâmica são adquiridas por sensores do tipo *Light Detection And Ranging* (LIDAR) durante uma determinada navegação. Esse tipo de nuvem de pontos tem como principal aplicação o mapeamento de ambientes externos para proporcionar autonomia de dispositivos robóticos ou mesmo veículos. A Figura 8(a) apresenta um sistema de captura montado sobre um automóvel para realizar o mapeamento de um ambiente urbano, onde o resultado do mapeamento pode ser visto na Figura 8(b). O *dataset* proposto por (PANDEY; MCBRIDE; EUSTICE, 2011) disponibiliza nuvens de pontos de mapeamentos de alguns ambientes através de aquisição dinâmica.

As nuvens de pontos de aquisição dinâmicas são classificadas como de Categoria 3 pelo MPEG. Esse tipo de nuvens de pontos tem como codificador o software G-PCC. As nuvens de pontos de aquisição dinâmica que contemplam as CTC, são disponibilizadas em arquivos no formato PLY como as nuvens de pontos das outras categorias apresentadas. A nuvem do tipo aquisição dinâmica é uma nuvem estática, ou seja, é constituída de um único arquivo para mapeamento de um cenário completo. Nesse contexto, soluções para compressão desse tipo de conteúdo tendem a explorar redundância espacial na codificação, como na codificação de nuvens estáticas. Dessa forma, o MPEG propõe uma única solução para compressão que contempla as duas categorias, categoria 1 e 3 (SCHWARZ et al, 2018).

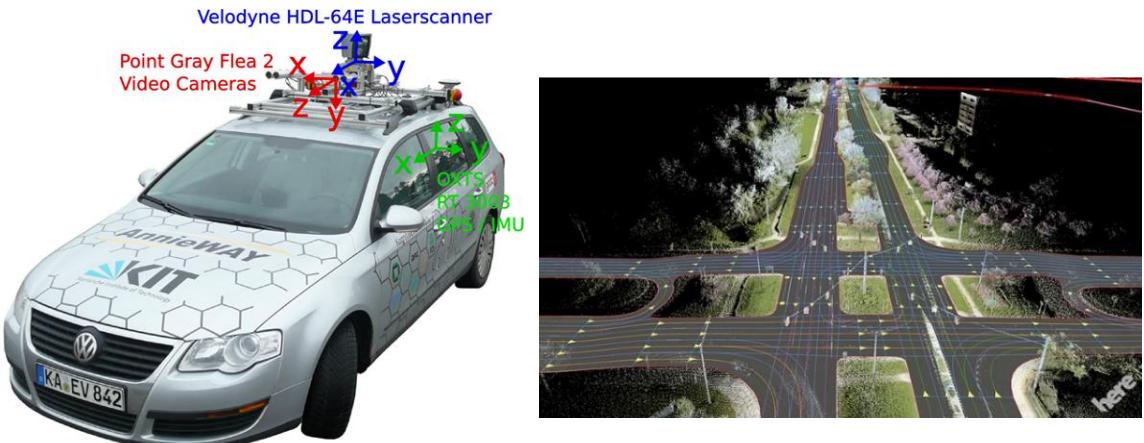


Figura 8: Sistema de aquisição das nuvens de pontos (a) e nuvens de pontos geradas (b) (TULVAN; MEKURIA; LI, 2016).

## 2.2 Redundância de Informação em Nuvens de Pontos

De modo geral, a compressão busca diminuir a quantidade de dados considerados redundantes ou irrelevantes na representação das informações, tais como textos, áudios, imagens, vídeos, entre outros. Dessa forma, é possível transmitir este tipo de informação com uma menor taxa de bits e/ou ainda armazenar a informação com uma menor demanda de espaço de armazenamento. Considera-se redundante aquele dado que não contribui (ou contribui muito pouco) para a representação da informação. As redundâncias presentes nos vídeos digitais, e que também podem ser encontradas nas nuvens de pontos, podem ser divididas basicamente em três diferentes tipos:

- Redundância Espacial – A redundância espacial, também chamada de redundância intra-quadro advém da correlação existente entre os pontos espacialmente distribuídos em um quadro ou uma nuvem de pontos.
- Redundância Temporal – A redundância temporal, também chamada de redundância inter-quadros ou inter-nuvens, é causada pela correlação existente entre nuvens temporalmente vizinhas. Muitos blocos de pontos de nuvens temporalmente vizinhas permanecem estáticos como por exemplo, um fundo que não é alterado em uma sequência de nuvens. Outros pontos apresentam uma pequena alteração causada, por exemplo, por uma variação de iluminação. Por fim, também é possível que o bloco de pontos simplesmente tenha se deslocado em relação a uma nuvem temporalmente

vizinha, como em um movimento de um objeto de uma cena. A exploração eficiente da redundância temporal conduz a elevadas taxas de compressão, o que é fundamental para o sucesso dos codificadores.

- Redundância Entrópica – A redundância entrópica está relacionada com as probabilidades de ocorrência dos símbolos. A entropia é uma medida da quantidade média de informação transmitida por símbolo (SHI; SUN, 1999). A quantidade de informação nova transmitida por um símbolo diminui na medida em que a probabilidade de ocorrência deste símbolo aumenta. Então, os codificadores que exploram a redundância entrópica têm por objetivo transmitir o máximo de informação possível por símbolo codificado e, desse modo, representar mais informações com um número menor de bits. Isto é feito a partir da atribuição de códigos menores (menos bits) para símbolos de maior ocorrência e códigos maiores para símbolos de menor ocorrência, ao invés do uso de um número fixo de bits para qualquer símbolo.

Neste sentido, este trabalho apresenta uma solução que trata da correlação temporal por meio de uma ME que considera a informação tridimensional das nuvens de pontos. Além disso, este trabalho também propõe um subparticionamento de blocos que contribui com uma predição intra mais flexível, oferecendo a possibilidade de refinar regiões mais complexas, como regiões com mais texturas/bordas, por meio de blocos de tamanhos menores. Além disso, também é apresentado um algoritmo para a decisão de modo, que escolhe entre a melhor predição inter/intra, além de definir a melhor opção de particionamento para cada macrobloco codificado.

### 2.3 Compressão com Perdas e sem Perdas

Os codificadores de nuvens de pontos podem ser classificados entre os que fazem a compressão sem perdas (*lossless*) e os que geram perdas durante o processo de compressão (*lossy*) (SCHWARZ; CHOU; SINHAROY, 2018). Na compressão sem perdas, os codificadores utilizam apenas técnicas de compressão que não eliminam informações, garantindo que o arquivo resultante do processo de decodificação seja idêntico ao arquivo original. Esse tipo de compressão pode ser muito eficiente para diferentes tipos de dados, principalmente para os que apresentam grande quantidade de redundância. Um exemplo de compressão sem perdas muito utilizado é o

compressor de arquivos ZIP (SALOMON, 2008). Esse algoritmo reestrutura o arquivo atribuindo códigos menores para símbolos com maior ocorrência e códigos maiores para símbolos de menor ocorrência, reduzindo assim o tamanho do arquivo sem perda de informação. No entanto, imagens e vídeos digitais possuem uma distribuição de probabilidade muito próxima entre os símbolos (valores dos pixels). Compressores de imagens sem perdas atingem limitadas taxas de compressão e o mesmo ocorre em vídeos (GHANBARI, 2017). Além da relativa ineficiência dos compressores sem perdas neste contexto, o volume de dados a ser processado, armazenado ou transmitido, principalmente em vídeos é muito elevado e requer taxas de compressão maiores. A compressão com perdas é a mais utilizada neste tipo de aplicação.

A compressão sem perda é importante, por exemplo, para conteúdos como imagens/vídeos para aplicações na área de medicina. Já em nuvens de pontos a compressão sem perda tem sido aplicada em mapeamentos de artefatos históricos. Em ambas aplicações é importante preservar os detalhes originais do conteúdo diante da importância desses (SCHWARZ; CHOU; SINHAROY, 2018).

Na compressão com perdas as informações de menor relevância podem ser eliminadas. Assim, é possível obter elevadas taxas de compressão com pouca redução de qualidade visual. Um exemplo é a operação chamada de subamostragem de cores, que é realizada sobre o espaço de cores YCbCr (componente luminância (Y), crominância azul (Cb) e crominância vermelha (Cr)) e aplicada nos codificadores de vídeo. Este processo consiste em reduzir a taxa de amostragem dos componentes crominância em relação ao de luminância, diante do fato do sistema visual humano ser mais sensível à informação de luminância do que às informações de crominância.

A compressão com perdas é a forma mais comum, pois apresenta taxas de compressão muito superiores às obtidas com a compressão sem perdas. Além disso, a grande maioria das aplicações de áudio e vídeo, de streaming, como por exemplo, vídeo conferência, são tolerantes a pequenas perdas de qualidade onde a taxa/s e volume de armazenamento são mais relevantes.

## 2.4 Octree

A decomposição ou particionamento através de *octree* é uma abordagem muito utilizada no âmbito de compressão de nuvens de pontos, especialmente se tratando

de métodos que exploram compressão na perspectiva de codificação intra-nuvem. A decomposição por meio de *octree* possibilita representar uma nuvem de pontos de forma mais compacta através de uma árvore constituída de dados binários. Uma *octree* é uma estrutura de dados em árvore em que cada nó contém oito filhos, também denominados octantes. As *octree* são frequentemente usadas para particionar espaços tridimensionais através de subdivisões recursivas (SCHNABEL; KLEIN, 2006).

No exemplo da Figura 9, é mostrada uma *octree* com seis níveis de decomposição, iniciando da esquerda (parte superior) até a direita (parte inferior) da Figura 9. Em cada nível da árvore, o espaço é subdividido por um fator 2, o que resulta em um incremento da resolução de voxels. A resolução pode ser especificada como critério de parada da decomposição, também chamado nível de detalhe (*Level of Detail – LoD*) da *octree*. No exemplo ilustrado na Figura 9, o nível seis é o último nível e, nesse caso, a *octree* tem *LoD* de 6 bits, ou ainda, resolução de  $2^6$  voxels em cada eixo (X, Y, Z).

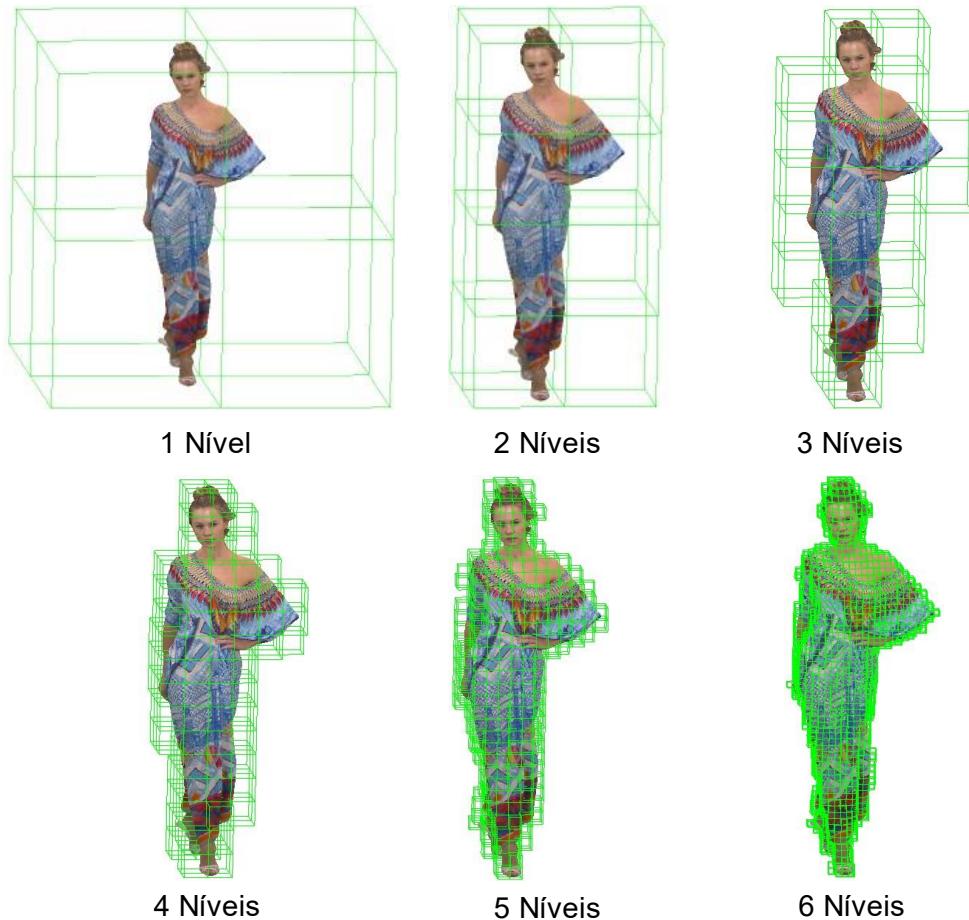


Figura 9: Seis níveis de octree aplicado na nuvem *Long Dress* (D'EON et al., 2016).

Dessa forma, a nuvem de pontos pode ser subamostrada preservando o ponto central de cada *voxel*. O ponto central em uma octante pode ser obtido por meio do cálculo do ponto médio, ou ainda, pode ser representado pela coordenada do centro da octante. Além disso, a *octree* pode ser representada de forma compacta, a partir de uma serialização, que pode ser chamada de mapa de ocupação.

Na Figura 10, um conjunto de bits pode ser utilizado para representar uma *octree*, sendo assim, o particionamento de cada nó é representado por um byte, como exemplificado pela raiz (“00000100”), onde o *voxel* ou octante que possuir pontos é representado pelo bit de valor 1 e o que não possuir é representado pelo valor 0. Na Figura 10 (lado direito) é ilustrada a representação de forma serializada (mapa de ocupação) da *octree*.

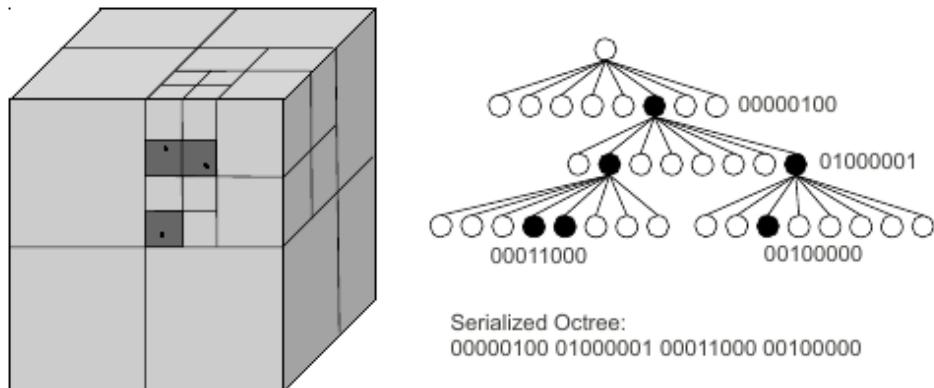


Figura 10: Esquema de serialização da *octree* (Figura adaptada de (Kammerl, 2012)).

O conceito de *octree* é amplamente aplicado para compressão de nuvens de pontos, devido à forma compacta de representação por meio dos mapas de ocupação, que favorecem a exploração de codificação entrópica (KAMMERL et al., 2012).

## 2.5 Avaliação de Qualidade de Nuvens de Pontos

Os algoritmos de compressão de nuvens de pontos, assim como os de vídeos 2D, exploram diferentes tipos de redundâncias para viabilizar menor demanda de armazenamento ou transmissão de dados. Porém, o resultado deste processo de compressão precisa manter qualidade satisfatória para visualização, mesmo que a exigência de qualidade possa variar dependendo do tipo de aplicação. Dessa forma, é preciso avaliar o impacto da compressão na qualidade das nuvens de pontos.

Os métodos de avaliação são classificados em duas categorias: subjetivos e objetivos. Os métodos subjetivos se baseiam na opinião de espectadores que devem observar as nuvens de pontos e responder um questionário elaborado de acordo com normativas de avaliação (MELO, 2017). Nesse sentido, a avaliação subjetiva demanda tempo, espaço físico e pessoas para sua aplicação. Os métodos objetivos utilizam equações calculadas por meio computacional (ALEXIOU; EBRAHIMI, 2017). Assim, tem como vantagem uma menor demanda de tempo e não necessitam pessoas e espaço físico. Além disso, as métricas de qualidade objetivas são indispensáveis diante da necessidade de os codificadores tomarem decisões sobre quais ferramentas devem ou não ser utilizadas durante o processo de codificação, tendo como base a avaliação de qualidade. Cabe destacar que o escopo desse trabalho se limita às métricas de qualidade objetivas.

A qualidade objetiva de nuvem de pontos é calculada para geometria e textura de forma separada. As métricas PSNR (*Peak Signal-to-Noise Ratio*) e MSE (*Mean Squared Error*), muito utilizadas em codificação de vídeo, são aplicadas para avaliar a qualidade tanto de geometria como de textura. Para a geometria existem duas formas mais comuns de calcular a qualidade, a abordagem ponto-a-ponto e a ponto-a-plano. Na ponto-a-ponto são calculadas as distâncias entre os pontos da nuvem original e os pontos mais próximos na nuvem reconstruída (após o processo de compressão e descompressão). No método ponto-a-plano, a qualidade é medida com base na distância entre cada ponto da nuvem reconstruída e um dado plano na nuvem original.

Nas próximas seções serão tratados maiores detalhes da aplicação dessas métricas para nuvens de pontos. Primeiramente serão apresentadas as métricas para avaliação de qualidade de geometria e posteriormente as métricas para textura.

### 2.5.1 Qualidade de Geometria

A avaliação de qualidade de geometria de acordo com documentos do MPEG possui duas principais formas de avaliação: ponto-a-ponto (*Point to Point* - P2P) e a ponto-a-plano (*Point to Plane* – P2Plane). A abordagem ponto-a-ponto é também denominada D1 e a abordagem ponto-a-plano denominada D2 (SCHWARZ; CHOU; SINHAROY, 2018). No entanto, neste trabalho serão utilizadas as denominações P2P e P2Plane.

Ambas as abordagens, P2P e P2Plane, são calculadas com a métrica MSE. O PSNR também pode ser calculado para obtenção do resultado final de qualidade. As métricas podem ser aplicadas considerando uma nuvem de pontos completa ou partes da nuvem, como por exemplo, blocos (SCHWARZ; CHOU; SINHAROY, 2018).

O cálculo da qualidade das nuvens de pontos após compressão considera uma nuvem de pontos original em relação a uma nuvem reconstruída (isto é, após o processo de compressão e descompressão). No entanto, na etapa de compressão com perda, os pontos são subamostrados, gerando uma redução do número de pontos da nuvem. Dessa forma, para que o MSE e o PSNR possam ser calculados é necessário obter o vizinho mais próximo (correspondente) para cada ponto da nuvem reconstruída na nuvem original. Somente através da obtenção dos pontos mais próximos é possível calcular as distâncias dos pontos (vetor de erro).

#### 2.5.1.1 Ponto-a-Ponto (P2P)

A abordagem ponto-a-ponto (*Point to Point* - P2P), calcula as distâncias entre os pontos da nuvem reconstruída e os pontos mais próximos na nuvem original. Para isso, é necessário primeiramente encontrar pontos mais próximos (correspondentes). Para que esses pontos sejam obtidos, é aplicada uma busca por meio do algoritmo *k-dimensional tree* (*Kd-tree*). A *Kd-tree* 3D assim como em vídeos 2D, é uma estrutura usada para organizar um número de pontos/pixels no espaço com *k* dimensões, e se caracteriza por ser uma árvore binária de pesquisa.

Na construção da *Kd-tree* 3D, para cada nível da árvore um eixo é selecionado para criar um plano de corte (particionamento do espaço 3D). O corte é baseado no ponto que corresponde à mediana dos pontos no eixo selecionado. A Figura 11 (b), ilustra um exemplo onde o nó raiz da árvore corresponde a mediana do eixo X, e consiste na criação do plano destacado na cor vermelha conforme ilustrado na Figura 11 (a). Os nós filhos dessa árvore consistem em planos de corte baseados no valor da mediana das coordenadas no eixo Y. Os nós netos são formados por planos no eixo Z e dessa forma a sequência se repete novamente até *K* dimensões. A partir do nó raiz, os pontos que possuem menor valor serão inseridos à esquerda da árvore e, consequentemente, os pontos de maior valor serão inseridos à direita.

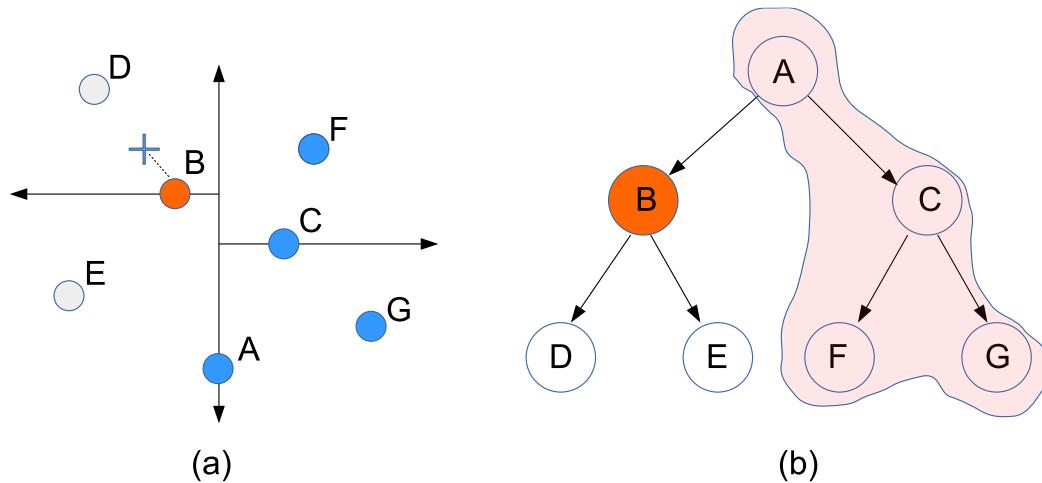


Figura 11: Exemplo de pesquisa do vizinho mais próximo na *Kd-tree*.

Para pesquisar o vizinho mais próximo de um dado ponto da nuvem de pontos reconstruída na nuvem de pontos original, são realizadas comparações entre o ponto que se pretende encontrar e os nós da árvore da nuvem original (SCHWARZ; CHOU; SINHAROY, 2018). Para isso, no início somente uma coordenada é considerada, por exemplo, a coordenada X. O procedimento varre apenas os nós que possivelmente levam ao encontro do ponto, tendo em vista que é uma árvore que possui valores ordenados. A busca termina ao encontrar um nó folha mais semelhante de valor igual ao da coordenada. Posteriormente, a métrica denominada quadrado da distância euclidiana (QE) é utilizada para calcular a distância entre o ponto encontrado e o ponto pesquisado (Sinal de “+”, Figura 11 (a)). O valor da distância é guardado juntamente com as coordenadas do ponto encontrado como o melhor candidato. A árvore é pesquisada novamente onde são avaliados o nó pai por meio da distância QE, assim como os nós folhas, como no exemplo ilustrado na Figura 11 (b) em que segue em destaque o nó B.

A partir da obtenção do ponto mais próximo para cada ponto da nuvem reconstruída B na nuvem original A, é possível calcular o vetor de erro  $E(i, j)$  conforme apresentado na Equação 1.

$$e_{B,A}^{D1}(i) = \|E(i, j)\|_2^2 \quad (1)$$

Na Equação 1, D1 representa o cálculo da distância por meio de P2P para cada ponto de B em A,  $E$  é o vetor de erro ou distância entre os pontos,  $i$  é a coordenada do ponto na nuvem B enquanto  $j$  é a coordenada do ponto mais próximo na nuvem A.

O cálculo da distância P2P para nuvem completa é realizado por meio da média da distância de todos os pontos, como apresenta a Equação 2.

$$e_{B,A}^{D1} = \frac{1}{N_B} \sum_{\forall i \in B} e_{B,A}^{D1}(i) \quad (2)$$

Na Equação 2, é calculada a média das distâncias de todos os pontos da nuvem B, onde  $N_B$  é o número de pontos da nuvem B. Essa média dos vetores de erro para a nuvem completa é utilizada para calcular o PSNR da nuvem de pontos como será descrito na Seção 2.5.1.3.

### 2.5.1.2 Ponto-a-Plano (P2Plane)

A abordagem ponto-a-plano (*Point to Plane* - P2Plane), também referida como (D2) pelo MPEG, mede a distância entre um ponto e um plano (modelo de superfície). Para calcular a métrica P2plane para a nuvem de pontos degradada B, é necessário encontrar o vizinho mais próximo na nuvem de pontos A. A etapa para calcular os pontos correspondentes entre as nuvens ocorre como na abordagem P2P. No entanto, para a abordagem P2plane o vetor de erro  $E(i, j)$  é projetado na direção do vetor normal  $N_j$  do ponto correspondente na nuvem A, gerando assim um novo vetor de erro  $\hat{E}(i, j)$ , como segue a Equação 3.

$$e_{B,A}^{D2}(i) = \|\hat{E}(i, j)\|_2^2 = (E(i, j) \cdot N_j)^2 \quad (3)$$

Na Equação (3),  $\hat{E}(i, j)$  é o novo vetor de erro calculado a partir do produto pelo vetor normal.  $N_j$  é o vetor normal para o ponto correspondente. O vetor normal ou superfície de um ponto é um vetor perpendicular ao plano tangente da superfície desse mesmo ponto. As normais em geral são dados intrínsecos incluídos nas nuvens de pontos utilizadas nas CTC propostas pelo MPEG.

A média dos vetores de erro  $\hat{E}(i, j)$  para todos os pontos é calculada como mostra a Equação 4. O resultado é utilizado para o cálculo do PSNR da nuvem completa.

$$e_{B,A}^{D2} = \frac{1}{N_B} \sum_{\forall b_i \in B} e_{B,A}^{D2}(i) \quad (4)$$

Na Equação 4, é calculada a média das distâncias de todos os pontos da nuvem B, onde  $N_B$  é o número de pontos da nuvem B e  $e_{B,A}^{D2}(i)$  é distância P2Plane de cada ponto.

A Figura 12 ilustra um exemplo de ponto da nuvem B (ponto preto) que tem como ponto mais próximo ou correspondente na nuvem A o ponto vermelho e, entre eles é calculado o vetor de erro. Posteriormente esse vetor é projetado como produto do vetor normal.

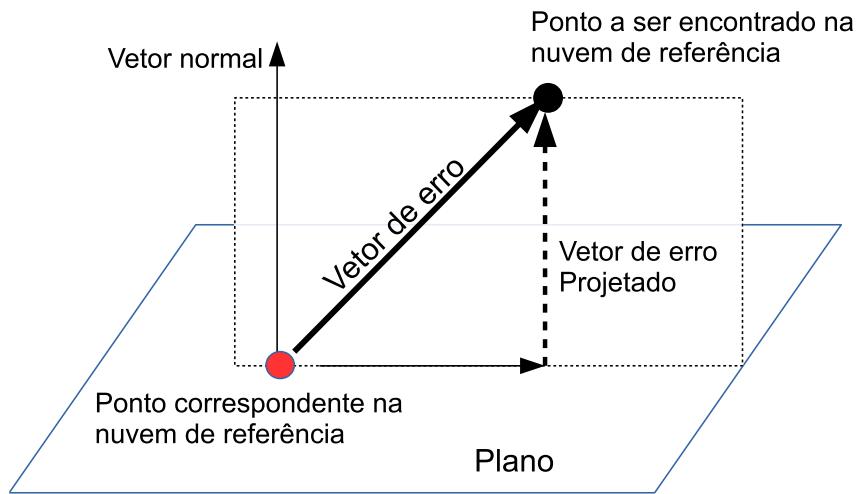


Figura 12: Exemplo da abordagem ponto-a-plano

O MPEG sugere ambas as abordagens, P2P e P2Plane, nas CTC. No entanto, conforme estudos apresentados em JAVAHERI (2017) e ALEXIOU (2017), a abordagem P2Plane apresenta resultados de qualidade mais correlacionados com a avaliação subjetiva, ou seja, essa é uma forma computacional de avaliar as nuvens de pontos mais fiel e aproximada de uma avaliação feita por meio da observação de pessoas. Conforme JAVAHERU (2017), as métricas P2Plane têm maior correlação com as avaliações subjetivas, uma vez que essas métricas modelam a superfície subjacente da nuvem de pontos e as distorções típicas para as nuvens de pontos reconstruídas correspondem a leves deslocamentos na superfície subjacente, que podem levar a erros significativos nas métricas do P2P, sem um impacto perceptivo significativo após a renderização. Neste sentido, neste trabalho serão apresentados os valores de ambas as abordagens, porém será dada maior relevância para resultados adquiridos por meio da P2Plane.

### 2.5.1.3 Métricas de similaridade

O cálculo do MSE para nuvens de pontos é a média das distâncias quadráticas dos pontos da nuvem degradada B e os vizinhos mais próximos da nuvem de pontos de referência A. Para o cálculo do vetor de erro é aplicada a distância quadrática dos pontos e posteriormente é calculada a média desses vetores de erro. Dessa forma, o MSE já é calculado tanto através da abordagem P2P como para P2plane, apresentadas nas seções 2.5.1.1 e 2.5.1.2, respectivamente. A equivalência entre as equações é apresentada na Equação 5 (SCHWARZ; CHOU; SINHAROY, 2018).

$$MSE = e_{B,A}^{D1} = \frac{1}{N_B} \sum_{\forall i \in B} e_{B,A}^{D1}(i) \quad (5)$$

Onde  $e_{B,A}^{D1}$  é a distância P2P média para os pontos da nuvem de pontos B em relação à nuvem A, calculada conforme apresentado na Equação 2. O MSE pode ser ainda calculado por meio da Equação 4 para obtenção de resultados para a abordagem P2Plane.

Os valores de PSNR são medidos em decibéis (dB), sendo que para nuvens idênticas que resultam em um valor de MSE igual a 0, o valor de PSNR será indefinido. Valores altos de PSNR indicam qualidade relativamente alta e valores baixos indicam qualidade relativamente baixa (SCHWARZ; CHOU; SINHAROY, 2018).

O PSNR é apresentado na Equação 6, onde  $p$  corresponde ao valor máximo que pode ser assumido por uma amostra. Para a nuvens de pontos dinâmicas o valor de  $p$  deve ser 1024 conforme estabelecido pela CTC (SCHWARZ; CHOU; SINHAROY, 2018).

$$PSNR = 10 \log_{10} \left( \frac{p^2}{MSE} \right) \quad (6)$$

A nuvem de pontos reconstruída por meio de compressão com perdas pode ter seus pontos subamostrados. Como resultado disso, pode haver uma diferença de número de pontos entre a nuvem reconstruída e a nuvem original. Para compensar estas diferenças, as abordagens P2P e P2plane devem ser calculados de forma simétrica. Nesse contexto, quando realizado o cálculo de P2P ou P2plane entre a nuvem reconstruída e a nuvem original, o MSE deve ser calculado entre a nuvem original e a reconstruída, e também na ordem contrária (SCHWARZ; CHOU;

SINHAROY, 2018). Isso porque para os pontos subamostrados devem existir pontos vizinhos mais próximos na nuvem de referência do que quando realizado o cálculo inverso (nuvem de referência em relação a reconstruída), em que os pontos próximos tendem estar mais distantes. Assim, o cálculo de forma simétrica tem como objetivo garantir uma maior robustez para a avaliação de qualidade. Nesse contexto, o PSNR utiliza o maior valor de MSE conforme a Equação 7.

$$PSNR = 10\log_{10}\left(\frac{p^2}{\max(e_{B,A}^{Dx}, e_{A,B}^{Dx})}\right) \quad (7)$$

Na Equação 7,  $p^2$  representa o pico de sinal e  $\max(e_{B,A}^{Dx}, e_{A,B}^{Dx})$  define o *MSE* simétrico, ou seja, deve-se utilizar o maior *MSE* entre a nuvem B e A e entre A e B.

### 2.5.2 Qualidade de Cor

A avaliação de qualidade de cor é baseada na métrica P2P, ou seja, para cada ponto da nuvem reconstruída são pesquisados os vizinhos mais próximos (correspondentes) na nuvem de referência. Para a obtenção dos pontos correspondentes entre as nuvens de pontos é considerada somente a informação de geometria dos pontos, ou seja, a distância entre os pontos. Com a obtenção dos pontos correspondentes para a nuvem reconstruída é calculada a qualidade da textura. O vetor de erro passa a ser baseado nos atributos de cor e a médias desses contemplam o MSE.

O cálculo dos atributos de cor para nuvens de pontos é realizado por meio do sistema de cor YUV. Assim, é necessário converter as informações do sistema RGB para YUV. O MSE e o PSNR são calculados de forma individualizada para cada canal, conforme SCHWARZ (2018).

Como feito para a avaliação de geometria, na avaliação da textura também é utilizado o MSE simétrico para calcular o PSNR. Segue de forma geral na Equação 8, o cálculo do PSNR para os atributos de cor.

$$PSNR = 10\log_{10}\left(\frac{p^2}{MSE_{simétrico}}\right), \quad (8)$$

Conforme a Equação 8,  $p^2$  representa o pico de sinal e  $MSE_{simétrico}$  é o maior MSE entre nuvem de referência e nuvem corrente.

## 2.6 Trabalhos relacionados

Nesta seção será apresentado um breve relato dos principais trabalhos encontrados na literatura que tratam da compressão de nuvens de pontos. Esses trabalhos podem ser classificados em duas classes, a que contempla exploração 3D, seja por meio de grafos ou árvores, e a que explora a projeção da informação 3D para 2D para realizar a compressão. Além disso, dentro dessas duas classes, existem ainda os trabalhos focados somente em compressão de geometria ou em compressão de textura, como também trabalhos que contribuem para compressão de ambas as informações. É interessante destacar que na literatura atual também são encontrados trabalhos que propõem codificadores completos que abordam tanto aspectos de predição intra como inter-nuvens e tratam informação de geometria e textura.

Como propostas de trabalhos que exploram 3D com a utilização de árvores, é possível destacar abordagens como (GARCIA; QUEIROZ, 2018), (SMITH; PETROVA; SCHAEFER, 2012) e (HUANG, 2006), em que é explorada somente informação de geometria. Propostas que exploram tanto geometria quanto textura são abordadas em (GOLLA; KLEIN, 2015), (HUANG, 2008), (SCHNABEL; KLEIN, 2006), e (QUEIROZ; CHOU, 2016).

Como trabalhos que propõem uso de grafos para compressão de geometria e textura se enquadram as propostas de (THANOU; CHOU; FROSSARD, 2015) e (THANOU; CHOU; FROSSARD, 2016). Abordagens que tratam somente textura são encontradas em (ZHANG; FLORENCIO; LOOP, 2014), (ZHANG; FLORÊNCIO, 2012) e (SHAO et al, 2017).

No grupo de abordagens que propõem codificar nuvens de pontos por meio de projeções 2D são encontradas propostas de soluções em (MORELL et al., 2014) e (LOOP; ZHANG; ZHANG, 2013).

A exploração da predição inter-nuvens, ou seja, para a categoria de nuvens de pontos dinâmicas, utilizando de técnicas que exploram 3D com a aplicação de árvores como, por exemplo, *octree*, são encontradas em (GARCIA; QUEIROZ, 2017) e (QUEIROZ; CHOU, 2017).

Soluções que propõem codificadores completos são encontradas em (MAMMOU, 2018), (MEKURIA; BLOM; CESAR, 2017), (QUEIROZ; CHOU, 2016) e

(KAMMERL et al., 2012). Entre os codificadores completos é possível destacar as soluções adotadas pelo MPEG para constituir ferramentas de exploração e servir de base comparativa para novas abordagens. Essas soluções são tratadas em (MEKURIA; BLOM; CESAR, 2017) e (MAMMOU, 2018) e serão apresentadas com maiores detalhes no Capítulo 3. Além disso, no Capítulo 4 serão discutidos, com maior profundidade, os trabalhos que tratam especificamente da exploração dos modos de predição intra e inter-nuvens de pontos para nuvens de pontos dinâmicas.

### 3 CODIFICADORES MPEG

Neste capítulo serão apresentados dois dos mais importantes *codecs* (codificadores e decodificadores) de nuvens de pontos dinâmicas encontrados na literatura atual, o *Point Cloud Compression* (PCC-Anchor) (MEKURIA; BLOM; CESAR, 2017) e o *Video-Based Point Cloud Compression* (V-PCC) (MAMMOU, 2018). Esses codificadores foram abordagens adotadas pelo MPEG após algumas chamadas de propostas do grupo. A apresentação desses codificadores tem como objetivo descrever as principais técnicas utilizadas para comprimir nuvens de pontos dinâmicas, além de destacar características importantes das predições. O conhecimento dessas soluções viabiliza uma melhor compreensão dos conceitos fundamentais, bem como o levantamento de aspectos relevantes para a contribuição deste trabalho.

#### 3.1 *Point Cloud Compression* (PCC-âncora)

O *codec* proposto por (MEKURIA; BLOM; CESAR, 2017) aborda tanto aspectos de predição intra como inter-nuvens. Esse *codec* foi utilizado como base para avaliação e desenvolvimento de novas abordagens para compressão de nuvens de pontos pelo MPEG, sendo denominado MPEG-PCC (*Point Cloud Compression*) (MEKURIA; CESAR, 2016). Porém, neste trabalho, ele será referido como PCC-âncora.

A Figura 13 ilustra uma visão geral do fluxo de codificação do PCC-âncora. Aqui as nuvens de pontos são primeiramente normalizadas para que se obtenha um alinhamento das caixas delimitadores (caixa número 1 da Figura 13), posteriormente é realizada a decomposição por meio de *octree* (caixa número 2 da Figura 13) como apresentado na Seção 2.4. Para a nuvem de pontos de referência é aplicada a codificação entrópica sobre os mapas de ocupação para os dados de geometria gerados pela *octree* (caixa número 3 da Figura 13).

Na compressão dos atributos de cor é realizada uma varredura na árvore onde

os pontos são mapeados para uma matriz 2D e, posteriormente é codificada com o algoritmo JPEG clássico (WALLACE, 1992) (caixa número 4 da Figura 13). A nuvem corrente, ou seja, a segunda nuvem de entrada, sofre uma decomposição por meio de *octree*, com o objetivo de gerar macroblocos (caixa número 5 da Figura 13). Cada macrobloco deve ser avaliado se será codificado como intra ou inter mediante a avaliação de alguns critérios. Aos blocos selecionados para serem codificados como inter, é aplicada uma quantização para os vetores de movimento (caixa número 6 da Figura 13), conforme será explicado na Seção 3.1.1.2. Por outro lado, os blocos que serão codificados como intra seguem o mesmo caminho de codificação que a nuvem de referência. Ao final é realizada a geração do cabeçalho e codificação entrópica para o *bitstream* (caixa número 7 da Figura 13).

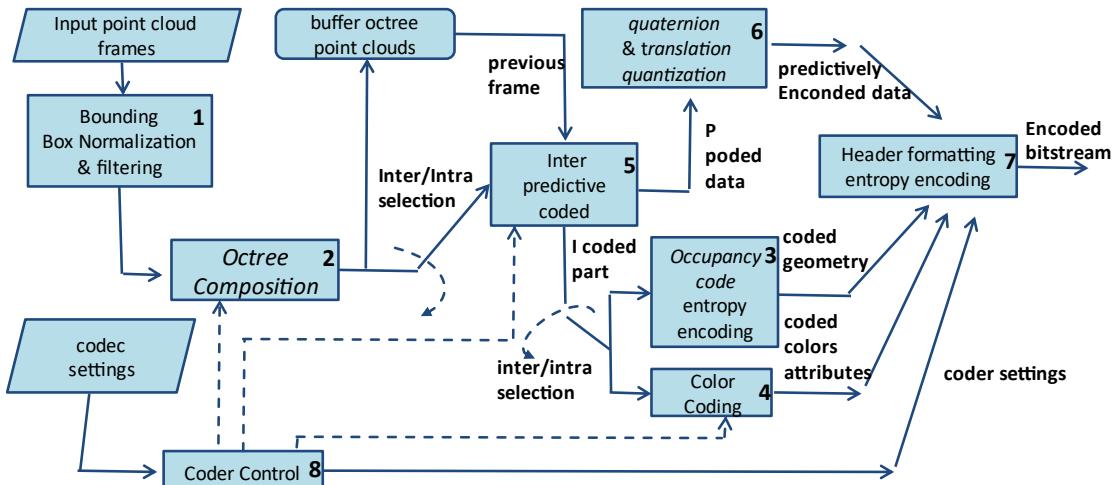


Figura 13: Visão geral do codec PCC-âncora (adaptado de (MEKURIA; BLOM; CESAR, 2017)).

### 3.1.1 Predição Inter-nuvens

Na etapa de predição inter-nuvens, primeiramente é realizada uma normalização das coordenadas das nuvens de pontos. Essa normalização também tem função de alinhar as nuvens de pontos, pois insere as nuvens temporalmente vizinhas em intervalos conhecidos e iguais, isto é, intervalo 0 e 1 (MEKURIA; BLOM; CESAR, 2017). O alinhamento é importante devido às características de irregularidade dos tamanhos de nuvens temporalmente vizinhas. Nesse sentido, para calcular a normalização são obtidas as coordenadas mínimas e máximas nos três eixos considerando as duas nuvens. Obtendo esses limites de cada eixo, os valores são usados para

normalizar as nuvens de pontos (MEKURIA; BLOM; CESAR, 2017). A Figura 14, mostra duas nuvens de pontos de entrada (parte superior da figura) e as nuvens de pontos após o alinhamento e normalização (parte inferior).



Figura 14: Normalização e alinhamento das caixas delimitadoras.

A predição inter-nuvens prevê a geração de blocos de tamanho  $N \times N \times N$  voxels, onde  $N$  é definido previamente como parâmetro de configuração do *codec*. Os resultados apresentados na proposta de (MEKURIA; BLOM; CESAR, 2017) utilizam  $N=16$ . Segundo o autor, esse tamanho de macrobloco habilitou os melhores resultados para os experimentos realizados.

Diferentemente do que ocorre na codificação de vídeos, nesse *codec* os blocos não são particionados em macroblocos menores, ou seja, uma vez definido o tamanho para os blocos, esses são inalteráveis. A ME é realizada a partir da comparação do macrobloco atual da nuvem corrente com o macrobloco colocalizado na nuvem temporalmente vizinha anterior (nuvem de referência).

### 3.1.1.1 Algoritmo ICP

O algoritmo ICP (*Iterative Closest Point*) é aplicado para calcular a matriz de translações e rotações para cada macrobloco, equivalente ao vetor de movimento produzido pelos codificadores de vídeo. Portanto, o algoritmo de ICP tem como objetivo estimar movimentos que um dado macrobloco possa ter sofrido na nuvem corrente visando minimizar a distância entre os dois blocos.

Para isso, o ICP tem como etapas:

1. Calcular correspondência dos pontos por meio do algoritmo *k-Nearest Neighbors (k-NN)* para selecionar os pontos vizinhos mais próximos;
2. Calcular a transformação para informação de geometria (rotação ou translação) e posteriormente é calculada a distância quadrática (*Square Distance*) entre um bloco da nuvem atual e a nuvem temporalmente anterior;
3. Aplicar a transformação (rotação ou translação) ao bloco se o resultado da distância quadrática respeitar um determinado limiar a ser definido;
4. Repetir todo o processo novamente.

Nesse sentido, o algoritmo realiza transformações (combinação de translação e rotação) necessárias para minimizar uma métrica de erro, ou seja, a distância do bloco atual e o bloco da nuvem de referência de forma iterativa. Para isso, primeiramente são selecionados alguns pontos correspondentes entre os blocos. A seleção desses pontos pode ser baseada na avaliação de características ou por meio de uma amostragem aleatória.

É importante destacar que o ICP estima o movimento do bloco considerando somente a informação de geometria dos pontos. Dessa forma, não é realizada avaliação de similaridade de textura dos pontos que compõem os macroblocos.

Como critério de parada do ICP são utilizados o número máximo de iterações, o valor máximo permitido para rotação e translação e o limiar para a distância quadrática, também chamado de *fitness score*. No trabalho proposto em (MEKURIA; BLOM; CESAR, 2017) foi utilizado um limite de 20 iterações e como valor de *fitness score* foi calculado o produto do *LoD* da *octree* em bits por quatro, ou seja, quatro vezes a quantidade de bits da representação *octree*.

### 3.1.1.2 Fluxo da Predição Inter-Nuvens

A matriz calculada pelo ICP é quantizada. Para isso, primeiramente são calculados os quaterniões (ADAMI; TREGLIA, 2002) para os valores de rotação. O cálculo de quaterniões consiste basicamente em gerar uma representação mais compacta da matriz de rotação, em que a matriz de 3x3 é representada por um vetor de 4 componentes. Posteriormente, é realizada uma quantização que converte os quatro componentes dos quaterniões em três valores. Assim, ao final da quantização dos quaterniões, os nove valores da matriz referentes às rotações são convertidos em um vetor de três posições. Ao final do processo de quantização, o vetor de movimento é constituído por três componentes de translação, três componentes de rotação e três componentes das coordenadas do macrobloco na nuvem de referência, também chamadas de *keys*. A Tabela I apresenta a estrutura do vetor de movimento para codificação inter-nuvens proposta por (MEKURIA; BLOM; CESAR, 2017). Cada componente é armazenado em uma variável inteira de 16 bits, assim cada linha da Tabela I, utiliza 6 bytes, totalizando 18 bytes o vetor completo.

Tabela I: A estrutura dos dados que compõem o vetor de movimento.

Coordenada X	Coordenada Y	Coordenada Z
Quaternião 1	Quaternião 2	Quaternião 3
Translação X	Translação Y	Translação Z

A utilização das *keys* é justificada pela possibilidade de exploração de paralelismo no processo de decodificação e as características inerentes as nuvens de

pontos como a falta de regularidade de voxels ocupados, conforme descrito em (MEKURIA; BLOM; CESAR, 2017).

### 3.1.1.3 Decisão de Modo

A decisão do melhor modo de predição, inter ou intra-nuvem, segue um conjunto de testes com seus respectivos critérios:

- **Número de pontos:** o macrobloco corrente não deve ter um número de pontos menor que cinquenta por cento dos pontos da nuvem de referência, nem mesmo pode ter um número de pontos cinquenta por cento maior que os pontos da nuvem de referência.
- **Variância de cor:** para cada macrobloco é calculada a variância de cor. Segundo (MEKURIA; BLOM; CESAR, 2017), macroblocos codificados como inter com alta variância de cor podem gerar artefatos visíveis em regiões da nuvem.
- **Convergência do ICP:** É verificado se algoritmo ICP obteve sucesso no cálculo da matriz de rotações e translações e, posteriormente o resultado é verificado por meio de um nível de aptidão alcançado (*fitness score*).

O *fitness score* é calculado por meio da média das distâncias quadráticas do macrobloco do bloco da nuvem de referência em relação ao macrobloco da nuvem corrente. Para isso, através da matriz gerada pelo ICP é aplicada a transformação do bloco na nuvem de referência e posteriormente é avaliada a distância em relação ao macrobloco corrente. O valor da distância é confrontado com o *fitness score* definido por (MEKURIA; BLOM; CESAR, 2017), que tem como valor a resolução dos pontos após subamostragem por *octree*, multiplicado por quatro. Dessa forma, se o valor estiver abaixo do *fitness score*, o bloco será codificado como inter; caso contrário, o macrobloco será codificado como intra.

A predição inter-nuvens corresponde ao bloco seis da Figura 15 e o seletor se refere aos critérios citados (bloco cinco da Figura 15). O fluxo é melhor representado através da Figura 15, primeiramente são gerados macroblocos para a nuvem de referência e nuvem corrente (bloco número 1). Para cada macrobloco da nuvem corrente é verificada a existência de pontos na nuvem de referência (bloco número 2 e 3 da Figura 15). Além disso, é realizada a verificação do número de pontos conforme critérios apresentados (bloco 4), assim como a variância de cor (bloco 5). O ICP é

calculado (bloco 6) e é verificada a convergência e o *fitness score* para o macrobloco (bloco de número 7 e 8). Se todos os critérios forem satisfeitos para codificar o macrobloco como inter, a matriz de rotação e translação é codificada (bloco 9), ou seja, são calculados os quaterniões e é aplicada uma quantização sobre os componentes, conforme já descrito. Dessa forma, é enviado ao *bitstream* a coordenada do macrobloco (*key*), a matriz de rotação e translação do macrobloco já codificada (*rigid tf*). Caso algum dos critérios não seja satisfeito, o macrobloco é codificado como intra.

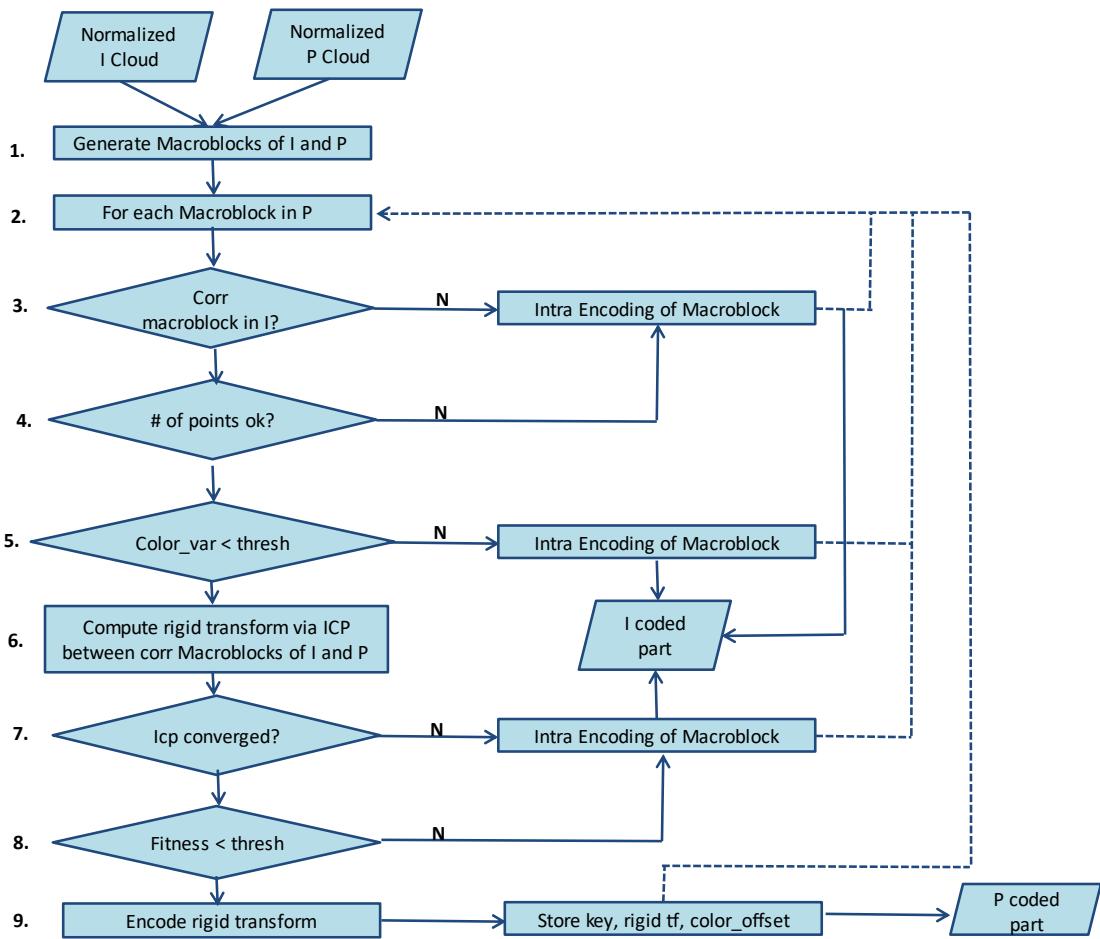


Figura 15: Fluxo da predição inter-nuvens e decisão de modo (adaptado de (MEKURIA; BLOM; CESAR, 2017)).

### 3.1.2 Predição Intra-nuvens

Para a etapa de predição intra-nuvem é realizado o processo de decomposição por *octree* (KAMMERL et al., 2012). Os mapas de ocupação gerados são enviados para o codificador entrópico proposto por (HUANG et al., 2006). Para a codificação de atributos de textura, é calculada a média de todos os pontos pertencentes à região

que contempla um *voxel* gerado da decomposição por *octree*. Posteriormente, é realizada uma busca em profundidade na árvore para mapear todos os atributos de cor e inseri-los em uma matriz para construir uma imagem 2D. Através dessa imagem, a textura é codificada com o algoritmo JPEG. Na Figura 13, os blocos dois, três e quatro se referem às etapas da predição intra-nuvem.

A codificação intra para nuvem de pontos corrente é realizada por meio da união de todos os pontos pertencentes aos macroblocos que não atenderam aos critérios para predição inter. Esses pontos formam uma nova nuvem que é codificada da mesma forma que a nuvem intra de referência, ou seja, através da decomposição por *octree* para a geometria e a aplicação do JPEG clássico para os atributos de cor.

### 3.2 *Video-Based Point Cloud Compression (V-PCC)*

O V-PCC (*Video-based Point Cloud Compression*) foi o *codec* âncora/base de teste selecionado pelo MPEG após a chamada de propostas realizada em outubro de 2017 (SCHWARZ et al., 2018). O V-PCC apresenta uma abordagem baseada em vídeo, onde as nuvens de pontos passam por um processo de planificação e então são comprimidas pelo codificador de vídeo 2D HEVC (SULLIVAN et al., 2012). O objetivo do V-PCC é explorar as elevadas taxas de compressão obtidas com o HEVC para comprimir tanto as informações de geometria quanto de textura de nuvens de pontos dinâmicas. Esse também é referenciado na literatura como TMC2 (*Test Model Category 2*), conforme sugerido pelos documentos do MPEG (MAMMOU, 2018).

O processo de compressão inicia com a conversão da nuvem de pontos em duas sequências de imagens planificadas, uma para geometria e outra para textura (MAMMOU, 2018). Adicionalmente, metadados que são necessários para interpretar as duas sequências, isto é, os códigos de ocupação e informações auxiliares também são gerados e comprimidos separadamente. O *bitstream* gerado e os metadados são multiplexados para gerar o *bitstream* final, como segue ilustrado na Figura 16.

Na Figura 16, é ilustrada a estrutura do *bitstream* gerado pelo V-PCC, que mostra o grupo de quadros (*Group of Frames* - GOF) podendo ser chamado também de grupo de nuvens (*Group of Clouds* - GOC), composto pelo vídeo de geometria, informações auxiliares, códigos/mapas de ocupação e vídeo de textura (MAMMOU, 2018).

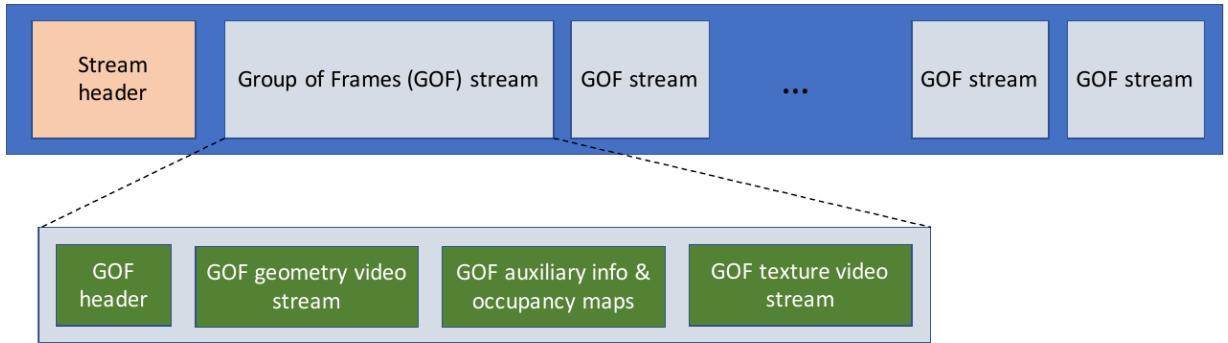


Figura 16: Estrutura do bitstream (MAMMOU, 2018).

Uma visão geral do processo de codificação pode ser vista na Figura 17. As maiores inovações desse *codec* estão nas etapas de geração de fragmentos (*Patch generation*), empacotamento (*Packing*) e o preenchimento (*Padding*). Essas etapas têm basicamente como objetivo tratar da geração de sequências de imagens 2D a partir das nuvens de pontos. Sequências distintas para textura e geometria são construídas, permitindo codificá-las em fluxos separados pelo HEVC (etapa denominada *video compression*) (MAMMOU, 2018).

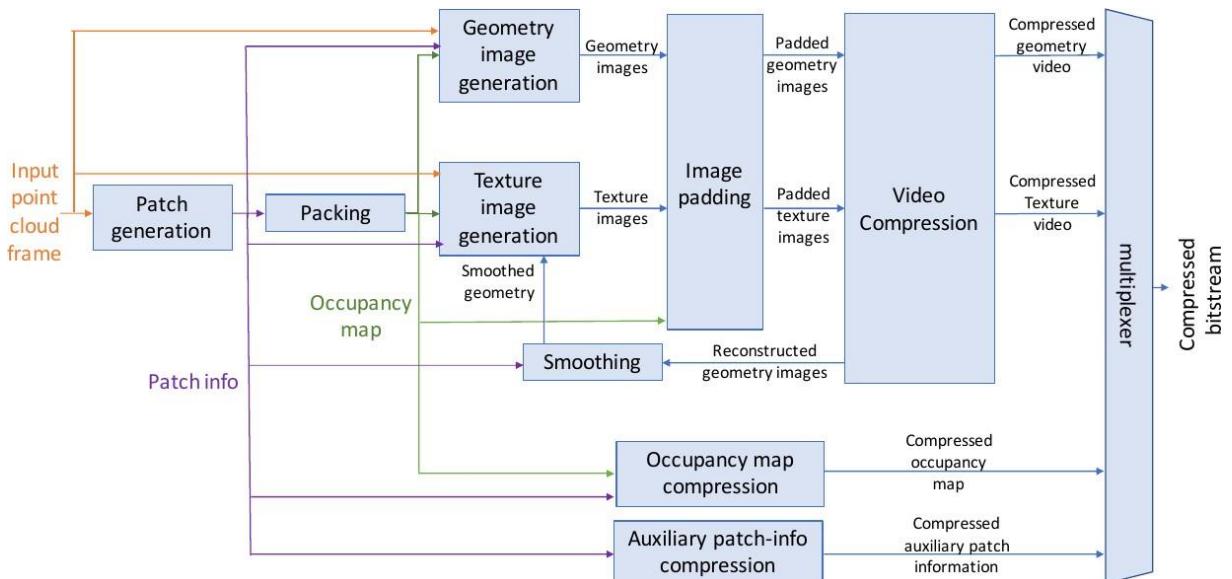


Figura 17: Visão geral do processo de codificação do V-PCC (MAMMOU, 2018).

A Figura 18 ilustra a visão geral do processo de decodificação do V-PCC. Na decodificação, as informações dos mapas de ocupação, assim como as informações auxiliares são imprescindíveis para a reconstrução das nuvens de pontos, considerando que estes dois fluxos carregam as informações para reconstrução do 3D.

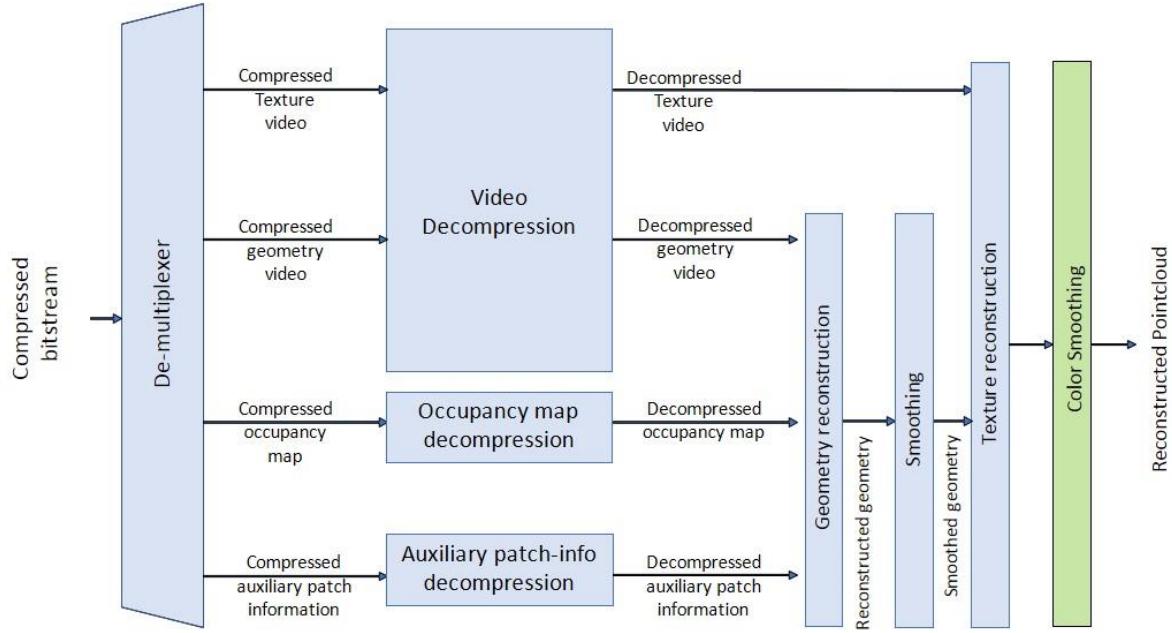


Figura 18: Visão geral do processo de decodificação do V-PCC (MAMMOU, 2018).

Para codificar as nuvens, o V-PCC prevê, primeiramente, a geração de fragmentos (*patches*) que são obtidos a partir do cálculo das normais de todos os pontos, conforme a abordagem apresentada em (HOPPE et al., 1992). O agrupamento é obtido associando cada ponto com um dos seis planos orientados, definidos por suas normais:  $(1.0, 0.0, 0.0)$ ,  $(0.0, 1.0, 0.0)$ ,  $(0.0, 0.0, 1.0)$ ,  $(-1.0, 0.0, 0.0)$ ,  $(0.0, -1.0, 0.0)$ , e  $(0.0, 0.0, -1.0)$ . Mais precisamente, cada ponto é associado ao plano que tem a normal mais próxima. Na Figura 19 são ilustradas as projeções desses fragmentos nos planos.

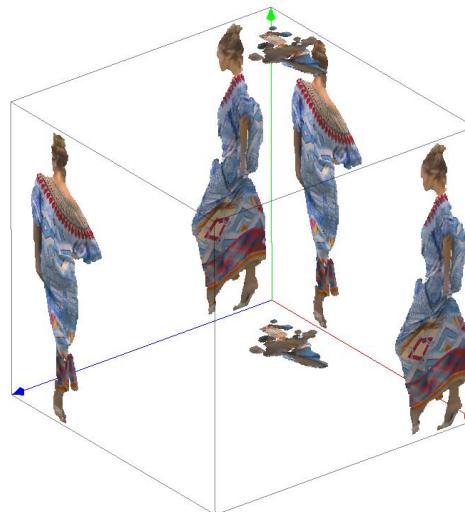


Figura 19: Nuvem de pontos projetada para os planos (ZAKHARCHENKO, 2018).

O processo de empacotamento (*Packing*) visa mapear os fragmentos extraídos em uma matriz 2D, enquanto tenta minimizar o espaço não utilizado garantindo que cada bloco  $T \times T$  (onde  $T$  é o tamanho do bloco, por exemplo,  $16 \times 16$ ) da matriz esteja associado a um único fragmento. Posteriormente, o processo de preenchimento (*Padding*) visa preencher o espaço vazio entre os *patches*, a fim de gerar uma imagem mais adequada para a compressão de vídeo. Entre os passos do preenchimento estão:

- Um processo independente para cada bloco  $T \times T$  (por exemplo  $16 \times 16$ );
- Se o bloco é vazio, então os pixels do bloco são preenchidos pela última linha ou coluna de bloco anterior;
- Se o bloco tiver pixels vazios e preenchidos, os pixels vazios serão preenchidos iterativamente com o valor médio de seus vizinhos não vazios.

O processo de geração de imagem explora o mapeamento 3D para 2D, calculado durante o processo de empacotamento, para armazenar a geometria e a textura da nuvem de pontos como imagem. Para lidar melhor com o caso de vários pontos que são projetados para o mesmo pixel, cada fragmento é projetado em duas imagens ou quadros. Basicamente, o processo consiste em inserir fragmentos que possuam maior profundidade para o quadro temporalmente sucessor, enquanto os fragmentos de menor profundidade devem ser inseridos no atual (MAMMOU, 2018). A Figura 20 ilustra uma nuvem de pontos após a etapa de geração de imagem.

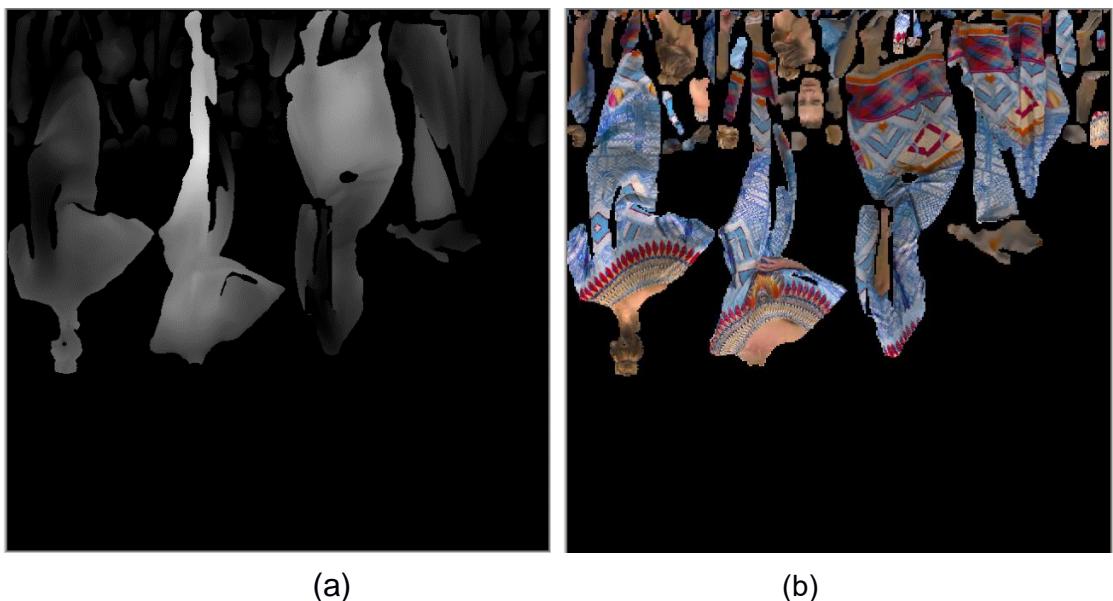


Figura 20: Exemplo de imagem de projeção para geometria (a) e cor (b) (ZAKHARCHENKO, 2018).

### 3.3 Considerações sobre os codificadores do MPEG

Os codificadores apresentados seguem abordagens distintas para realizar a compressão de nuvens de pontos dinâmicas. O PCC-âncora é focado em comprimir a informação de geometria por meio de *octree* e utiliza uma abordagem para predição inter-nuvens que explora redundância no espaço 3D. Por outro lado, o V-PCC utiliza uma abordagem baseada na conversão da nuvem de pontos em um vídeo 2D, usando o codificador de vídeo HEVC para comprimir tanto informação de geometria como textura. Nesse sentido, o V-PCC não explora as redundâncias temporais no espaço 3D original. A importância dessa exploração é comprovada diante da chamada para experimento de novas abordagens para ME 3D realizada pelo próprio MPEG, conforme documento (ZAKHARCHENKO; KIM, 2018). Uma abordagem recente, com foco na ME em 3D, foi proposta por (LI, et al., 2019) e implementada no codificador V-PCC. Porém, o método se limita a encontrar blocos colocalizados na nuvem de pontos 3D com o objetivo de auxiliar a ME 2D realizada no codificador HEVC. Para isso, a nuvem de pontos é reconstruída e a coordenada de um determinado *patch* 2D é convertida para 3D. Assim, por meio do bloco colocalizado, é possível obter a coordenada de um outro *patch* 2D que pode fornecer uma melhor correspondência. Essa abordagem tem como objetivo auxiliar a ME 2D, já que a divisão por *patch* pode inviabilizar a ME 2D, mesmo em blocos 3D espacialmente próximos ou colocalizados como pode ser visto na Figura 21.

Embora a proposta de (LI, et al., 2019) tenha como objetivo corrigir possíveis problemas da ausência de exploração do espaço 3D no V-PCC, essa abordagem ainda não explora, de forma satisfatória, as possibilidades em uma área de pesquisa tridimensional.

O V-PCC pode facilmente atingir menores taxas de bits de geometria na comparação com o PCC-âncora, no entanto, pode também apresentar resultados de qualidade de geometria inferiores (GONÇALVES, et al., 2019). Na Figura 22 são apresentados os gráficos de taxa e distorção (*Rate and Distortion - RD*) para a sequência *Loot* avaliada em (GONÇALVES, et al., 2019). Na Figura 22 são apresentados cinco *LoD*, representados pelos pontos das curvas, onde quanto maior o *LoD* maior a resolução das nuvens de pontos e consequentemente maior qualidade. Na Figura 22 (a) é possível ver que o PCC atinge níveis de qualidade de geometria mais altos para os *LoD*

maiores em comparação ao V-PCC, enquanto o V-PCC só atinge resultados satisfatórios para resoluções baixas.

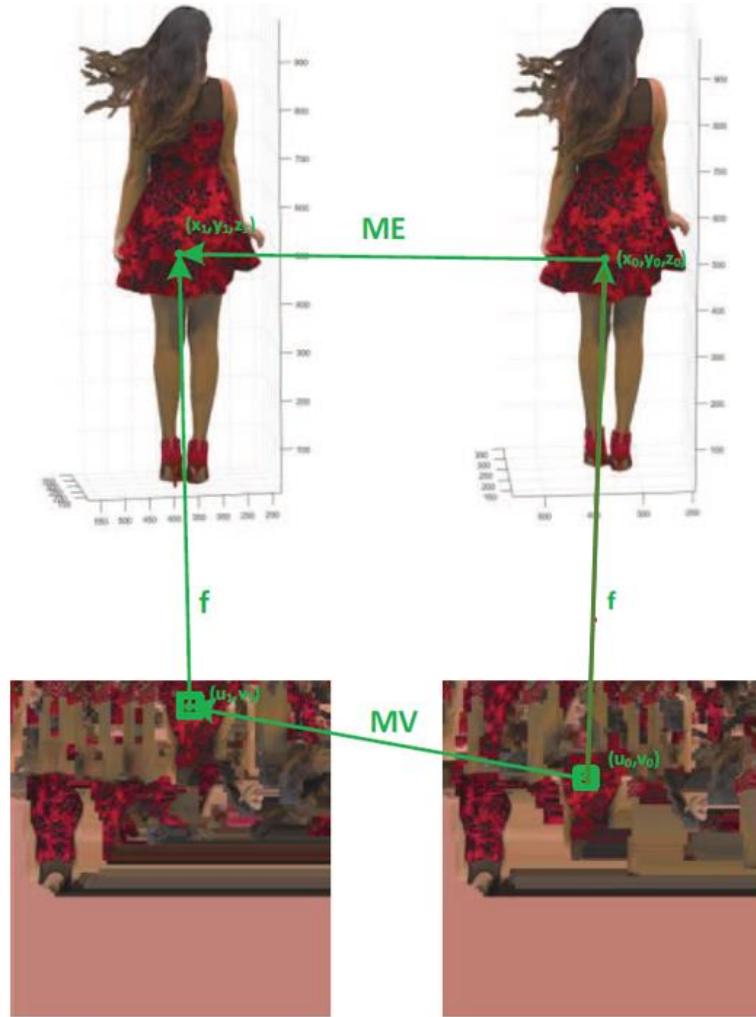


Figura 21: Problema na ME no V-PCC (adaptado de (LI, et al., 2019)).

Em relação à textura, os resultados obtidos pelo V-PCC são superiores em ambos os eixos, taxa e qualidade da nuvem codificada, como pode ser visto na Figura 22 (b) conforme apresentado em (GONÇALVES, et al., 2019). Estes resultados são esperados, uma vez que o V-PCC utiliza os algoritmos avançados do codificador HEVC, que é especializado na codificação de textura, enquanto o PCC utiliza um codificador JPEG para a compressão das informações de textura.

Neste trabalho, o codificador PCC-âncora será usado como base para o desenvolvimento das soluções propostas para a predição inter-nuvens no espaço 3D, visto que este codificador opera sobre os dados originais das nuvens de pontos, ou seja, no espaço 3D. Além disso, cabe destacar que a etapa de predição inter do V-PCC é exatamente a mesma utilizada no padrão HEVC, sem nenhuma inovação no cenário

da compressão de nuvens de pontos dinâmicas. É importante destacar também que a proposta desenvolvida nesta tese pode ser adaptada para o V-PCC em trabalhos futuros, diante da ausência desse tipo de solução e do potencial ganho de eficiência na predição que ela pode trazer.

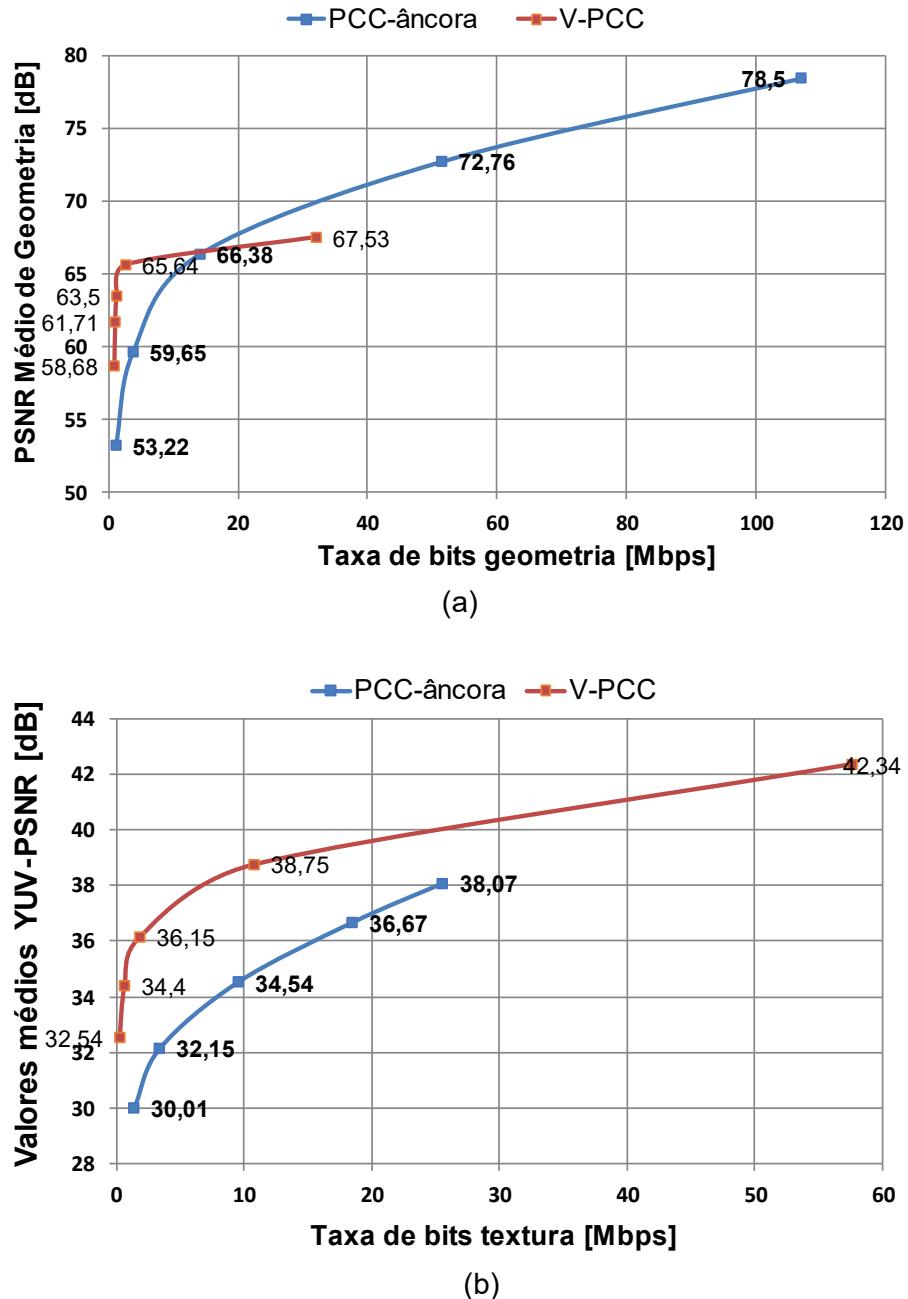


Figura 22: Comparativo entre V-PCC e PCC-âncora (adaptado de (GONÇALVES, et al., 2019))

## 4 PREDIÇÃO INTER-NUVENS

Nos compressores de vídeo, a eficiência de codificação é obtida por meio da exploração de redundâncias, tais como: redundância temporal, espacial e entrópica (SULLIVAN et al., 2012). As correlações temporais em nuvens de pontos dinâmicas ainda não são exploradas de forma satisfatória, e a eficiência da compressão não é significativamente aprimorada devido ao tamanho fixo de blocos, segundo Li (2019).

A ME é responsável por explorar a redundância temporal presente entre quadros temporalmente vizinhos em uma cena. Para isto, os quadros são divididos em blocos e uma busca é realizada comparando cada bloco do quadro atual (quadro sendo codificado) com blocos de um quadro de referência (previamente codificado). O bloco mais similar é selecionado e posteriormente é calculado o vetor de movimento, que contempla o deslocamento do bloco no quadro corrente em relação à posição de maior similaridade no quadro de referência. Para realização da busca pelo melhor bloco pode ser aplicado o algoritmo de busca completa (*Full Search – FS*) que apresenta o melhor resultado em termos de qualidade. O FS testa todas as possibilidades em uma determinada área de busca. Logo, sempre encontra o bloco de maior similaridade com o bloco do quadro atual. Porém, o custo computacional deste algoritmo pode, em muitos casos, ser proibitivo devido à grande quantidade de blocos candidatos que costumam estar presentes numa área de busca, especialmente quando aplicados à compressão de vídeos de alta resolução. Assim, na literatura são encontradas diferentes propostas de algoritmos rápidos de ME com o objetivo de reduzir o número de comparações realizadas e, consequentemente, reduzir o custo computacional da ME, mas mantendo a qualidade mais próxima possível do algoritmo FS.

Outro aspecto importante e tratado na predição inter-quadros em codificadores de vídeo como o H.264/AVC (*Advanced Video Coding*) e o HEVC (*High Efficiency Video Coding*) é a possibilidade do uso de blocos de tamanho variável durante a

predição (SULLIVAN; WIEGAND, 2005). Assim, macroblocos de tamanho 16x16 amostras, por exemplo, podem ser particionados em blocos de 8x8 e até mesmo subparticionados em blocos ainda menores, como 4x4. Os cálculos da estimação de movimento são realizados para vários tamanhos de blocos e assim a melhor combinação de tamanhos é escolhida. Essa escolha é realizada a partir da análise dos resultados de taxa (em bits) e distorção (qualidade objetiva) de cada tamanho de bloco visando sempre a opção que gere a codificação mais eficiente (SULLIVAN et al., 2012).

Assim como os avanços na etapa de predição inter-quadros trouxeram ganhos expressivos nas taxas de compressão dos compressores de vídeo, a exploração eficiente da ME em 3D pode viabilizar ganhos significativos para o aumento da eficiência de codificação de nuvens de pontos. Porém, devido ao acréscimo de uma dimensão em relação aos vídeos, a ME em 3D para nuvens de pontos apresenta uma complexidade ainda maior.

A variação dos tamanhos de blocos presente em codificadores de vídeos permite explorar, de maneira mais eficiente, as diferentes características de conteúdo presentes dentro dos quadros de um vídeo. Grandes regiões homogêneas e com pouca ou nenhuma variação podem ser mais eficientemente codificadas com blocos grandes. Já os blocos menores podem ser usados em regiões com maiores detalhes de textura, bordas e movimentos, auxiliando numa predição mais precisa. O uso de diversos tamanhos de bloco na ME para a codificação de nuvens de pontos pode, igualmente, beneficiar a compressão, principalmente porque uma das características das nuvens de pontos é a variação do número de pontos entre nuvens temporalmente vizinhas. Nesse contexto, a possibilidade o uso de blocos grandes, bem como do seu particionamento de diferentes tamanhos de bloco menores, pode viabilizar a seleção de blocos que possibilitem uma codificação com a melhor relação entre qualidade e taxa.

#### 4.1 Estado da Arte na Predição Inter-nuvens

As abordagens que tratam de nuvem de pontos dinâmica, também chamada de categoria 2 pelo MPEG, são soluções voltadas à correlação temporal (predição

inter-nuvens). Na literatura atual, essa categoria de nuvem de pontos ainda é pouco explorada.

O trabalho proposto em (KAMMERL et al., 2012) emprega uma *octree* modificada denominada *double octree*. Esta *double octree* é construída para aplicar uma operação de ou exclusivo sobre duas nuvens temporalmente vizinhas, reduzindo assim a entropia dos dados de geometria. No entanto, essa abordagem é limitada a nuvens de pontos que possuam pouco deslocamento entre nuvens temporalmente vizinhas.

O codificador proposto por (QUEIROZ; CHOU, 2017) explora correlação temporal com base em (DOU et al., 2015) que propõe uma malha triangular de pequenos conjuntos de pontos contíguos para constituir blocos. Cada bloco é codificado como intra ou inter-nuvens, por meio de uma decisão de modo que considera distorção de geometria e de textura, assim como taxa de bits, similarmente aos codificadores de vídeo. Para a codificação intra-nuvem de textura foi utilizada a abordagem chamada *Region-Adaptive Hierarchical Transform* (RAHT) (QUEIROZ; CHOU, 2016). Essa consiste em compactar informações de textura com base em uma transformação hierárquica usando *wavelet Haar*. Para a compactação da geometria, uma decomposição de *octree* é aplicada e posteriormente é aplicada uma codificação entrópica. Um algoritmo de filtragem baseado na morfologia matemática (operações de erosão/expansão) é proposto para suavizar aspectos de codificação.

O método proposto em (THANOU; CHOU; FROSSARD, 2015) realiza uma predição inter considerando informações de geometria e textura. As nuvens de pontos são representadas por uma sequência de grafos ponderados não direcionados. Para cada nuvem de pontos, um grafo ponderado é gerado e a correspondência entre os grafos é calculada. A predição inter considera a correspondência entre conjuntos de nós em cada grafo, que é primeiramente determinada por características esparsas usando o descriptor chamado *Spectral Graph Wavelet* (SGW). Para refinar o processo, uma predição inter-nuvens densa é realizada usando uma interpolação dos nós correspondentes. A compressão de textura é realizada também através da exploração de correlação temporal. Para isso, são calculados resíduos entre a nuvem de referência e a corrente, posteriormente é aplicada a transformada de Fourier, conforme proposta de (ZHANG; FLORÊNCIO, 2013). Os coeficientes de Fourier são quantizados enviados para o codificador entrópico proposto por (MALVAR, 2006).

Uma predição inter-nuvens baseada em árvore binária, denominada *k-dimensional tree (Kd-tree)* é proposta por (KATHARIYA et al., 2017). A *Kd-tree* possibilita um particionamento do espaço dependente dos dados, ou seja, divide o espaço em duas metades em cada nível. Cada metade contém o mesmo número de pontos, diferente do que realizado na decomposição por meio de *octree*. A estrutura em árvore é construída para a primeira nuvem de um determinado grupo de nuvens (GOC) e, assim, a mesma estrutura é procurada no restante das nuvens GOC. De acordo com (KATHARIYA et al., 2017), considerando que todas as nuvens GOC têm a mesma estrutura de árvore *Kd-tree*, as correspondências entre a nuvem de ponto de referência e a corrente são implícitas. A correspondência entre o bloco de nuvens de pontos atual e a referência é corrigida para obter uma correspondência individual entre os pontos. Isso é feito através da construção de um grafo bipartido em que o vizinho mais próximo na nuvem de referência é estimado. Embora esse trabalho apresente uma geração de macroblocos com base no número de pontos, as nuvens de pontos podem ter *voxels* ocupados ou não. Portanto, essa abordagem garante que o macrobloco selecionado tenha o mesmo número de pontos, mas pode não ser necessariamente o mais semelhante. Além disso, a correlação temporal é explorada avaliando apenas o bloco colocalizado.

A abordagem de (DOREA; QUEIROZ, 2018) propõe um método para reduzir o custo computacional para a predição inter-nuvens através da adição de informações para o mapa de ocupação, como tamanho do bloco, estatísticas do bloco local e número de *voxels* ocupados. Ao consultar o mapa é possível atingir uma redução significativa no espaço de pesquisa, evitando avaliações dispendiosas da correspondência de blocos. Para construir os mapas, descritores 3D são gerados através da contagem de *voxels* ocupados. Esse trabalho apresenta resultados de tempo e qualidade da estimação de movimento, sem apresentar resultados específicos para compressão como, por exemplo, o quanto efetivamente é possível atingir de redução de taxa de bits.

Nenhum dos trabalhos relacionados apresenta uma exploração da ME em 3D, nem mesmo o particionamento de blocos para refinar a ME. Conforme apresentado, os trabalhos encontrados na literatura consideram macroblocos colocalizados ou de tamanho fixo dentro de uma área de pesquisa. Diante disso, os aspectos mencionados no início deste capítulo apresentam novas possibilidades, ainda não exploradas, para a compressão de nuvem de pontos dinâmica. Nesse sentido, uma avaliação da

correlação temporal entre as nuvens temporalmente vizinhas, fundamental para uma exploração eficiente da ME em 3D, é apresentada na Seção 4.2.

## 4.2 Avaliação da Correlação Temporal em Nuvens de Pontos

Conforme apresentado nas Seções 2.6 e 4.1, as abordagens existentes, em geral, tratam somente blocos colocalizados na nuvem de referência ou, a exemplo do V-PCC, converte as nuvens para o espaço bidimensional, desprezando a exploração de correlação temporal no espaço 3D. Para isso, primeiramente é proposta uma avaliação da correlação temporal sobre informação de geometria das nuvens de pontos, a fim de obter um melhor entendimento das distribuições das redundâncias e permitir uma melhor exploração da predição inter-nuvens.

Para realizar essa avaliação, o algoritmo de estimação de movimento *Full Search* (FS) foi desenvolvido e aplicado para a informação de geometria das nuvens de pontos 3D. Na Figura 23 é apresentada a adaptação realizada do algoritmo FS para o espaço 3D. Para cada bloco da nuvem corrente é realizada uma pesquisa considerando uma área de busca (região em amarelo) ao redor do bloco colocalizado (bloco em azul) na nuvem de referência. Todos os blocos candidatos (com precisão de um ponto para cada eixo) são avaliados por meio do MSE. No entanto, para avaliar a similaridade de cada bloco primeiramente é calculado o vetor de movimento através do algoritmo ICP, conforme apresentado na Seção 3.1. Somente após a compensação de movimento do bloco candidato, é calculado o MSE. Assim, o bloco que tiver o menor valor de MSE é selecionado como a melhor correspondência.

Através da avaliação do algoritmo FS em 3D é possível construir um mapa de calor com as ocorrências das melhores correspondências na área de busca para todas nuvens que compõem uma sequência de nuvem dinâmica. Além disso, o mapa de calor também é importante para que seja descoberta a melhor configuração de tamanho de área de busca para encontrar a melhor correspondência. Neste sentido, o algoritmo FS foi avaliado com macroblocos de tamanho  $32 \times 32 \times 32$  voxels considerando para duas diferentes área de busca, sendo uma com  $[-7, +7]$  e outra com  $[-15, +15]$  voxels em cada eixo (X, Y, Z), gerando uma área de busca de  $46 \times 46 \times 46$  e  $62 \times 62 \times 62$  voxels, respectivamente.

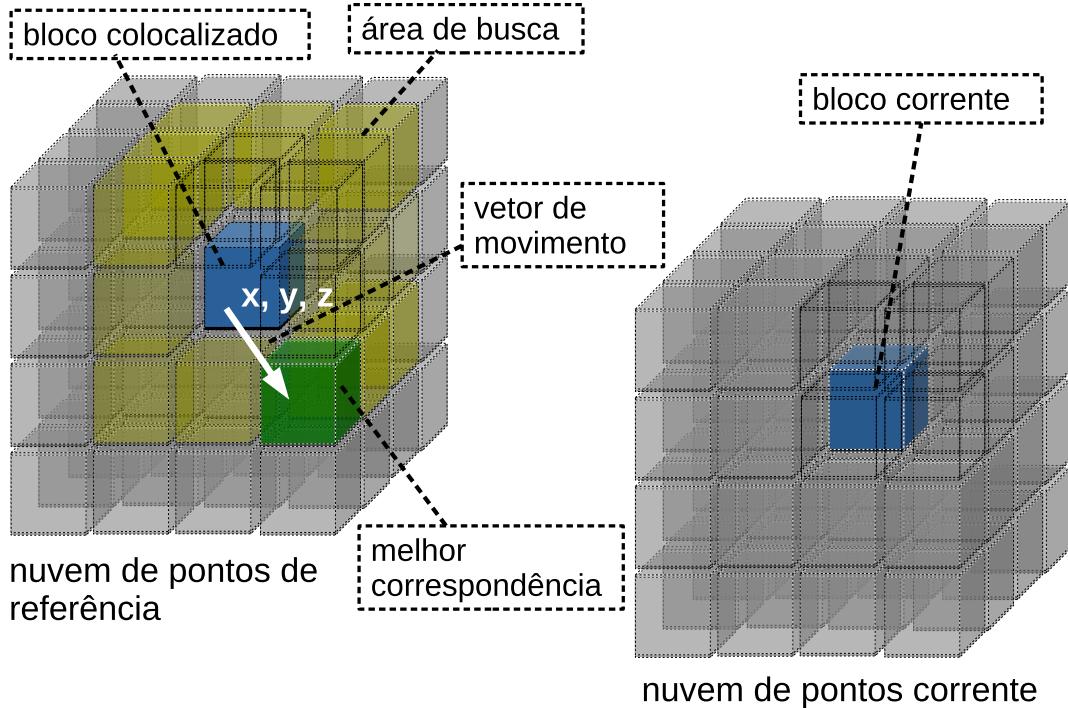


Figura 23: Visão geral da estimativa de movimento baseada em bloco para nuvem de pontos.

A Figura 24 ilustra os mapas de calor resultantes de quatro sequências de nuvem de pontos dinâmicas recomendadas pelas condições comuns de teste (*Common Test Conditions - CTC*) do MPEG (SCHWARZ; CHOU; SINHAROY, 2018). Esses mapas de calor foram construídos considerando as melhores correspondências do algoritmo FS para os primeiros 30 quadros de cada sequência de teste. Os pontos azuis escuros no mapa de calor representam menores ocorrências de melhor correspondência daquele determinado bloco candidato, enquanto pontos vermelhos representam maior número de ocorrências.

Como pode ser visto na Figura 24 (os mapas de calor do lado esquerdo, (a), (c), (e) e (g)), o intervalo de pesquisa  $[-7, +7]$  apresentou mapas de calor distribuídos para quase todas as regiões da nuvem. Somente a nuvem de pontos *Red and Black* teve a maioria de seus melhores casamentos concentrados na região central da área de pesquisa. Ao analisar esses resultados, é possível concluir que intervalos de pesquisa mais amplos devem ser usados, uma vez que diversos blocos candidatos com elevada ocorrência de escolha estão sendo encontrados próximo das bordas do intervalo de pesquisa  $[-7, +7]$ . Portanto, isto indica que pode ser possível encontrar correspondências ainda melhores além das bordas desse intervalo.

A Figura 24 também apresenta, no lado direito (b), (d), (f) e (h), os mapas de calor para o intervalo de pesquisa  $[-15, +15]$ . Nesse cenário, as ocorrências das

melhores correspondências estão mais concentradas no centro da área de busca, onde a maioria dos pontos vermelhos podem ser encontrados para todas as sequências de nuvens de pontos. No entanto, há também uma distribuição mais alta nos eixos X, Y e Z, que pode atingir a borda da área de pesquisa em alguns casos. Esse comportamento pode ser observado para a nuvem de pontos *Long Dress*, que apresenta algumas posições vermelhas no mapa de calor em todo o eixo Y até o limite da borda da área de pesquisa.

Essa avaliação mostra que os melhores resultados de similaridade entre os blocos de nuvens vizinhas podem, muitas vezes, estar localizados longe da região colocalizada. Portanto, isso destaca a necessidade de uma maneira mais eficiente de explorar a correlação temporal durante a codificação das nuvens de pontos do que a solução encontrada no PCC-âncora proposta em (MEKURIA; BLOM; CESAR, 2017). Além disso, os resultados apresentados mostram que um intervalo de pesquisa de [-15, +15] pode fornecer bons resultados para a ME, por fornecer uma área de busca mais adequada para a ME de nuvens de pontos dinâmicas. Portanto, esse intervalo será usado para as próximas avaliações neste trabalho.

Diante dos bons resultados apresentados pelo intervalo de pesquisa de [-15, +15] e também devido ao elevado tempo de simulação necessário, neste trabalho não foram exploradas áreas de busca maiores. Áreas de buscas menores também pode ser assunto para exploração futura, no entanto, como os resultados para intervalo de pesquisa de [-7, +7] mostram uma grande quantidade de ocorrência de seus melhores casamentos concentrados na região próximas aos limites da área de busca, optou-se, no primeiro momento, por desconsiderar avaliações em configurações de áreas de busca menores.

A avaliação da ME para diferentes tamanhos de macroblocos também pode ser interessante para que seja selecionada uma configuração que viabilize melhor eficiência de codificação. Assim, foi avaliada a ME para macroblocos de tamanho  $16 \times 16 \times 16$  voxels. Essas avaliações foram usadas com o objetivo de confrontar resultados de qualidade da ME para macroblocos de  $16 \times 16 \times 16$  voxels em relação a macroblocos de tamanho  $32 \times 32 \times 32$  voxels. Os mapas de calor para macroblocos de tamanho de  $16 \times 16 \times 16$  voxels podem ser visualizados no Apêndice A, Figura 47.

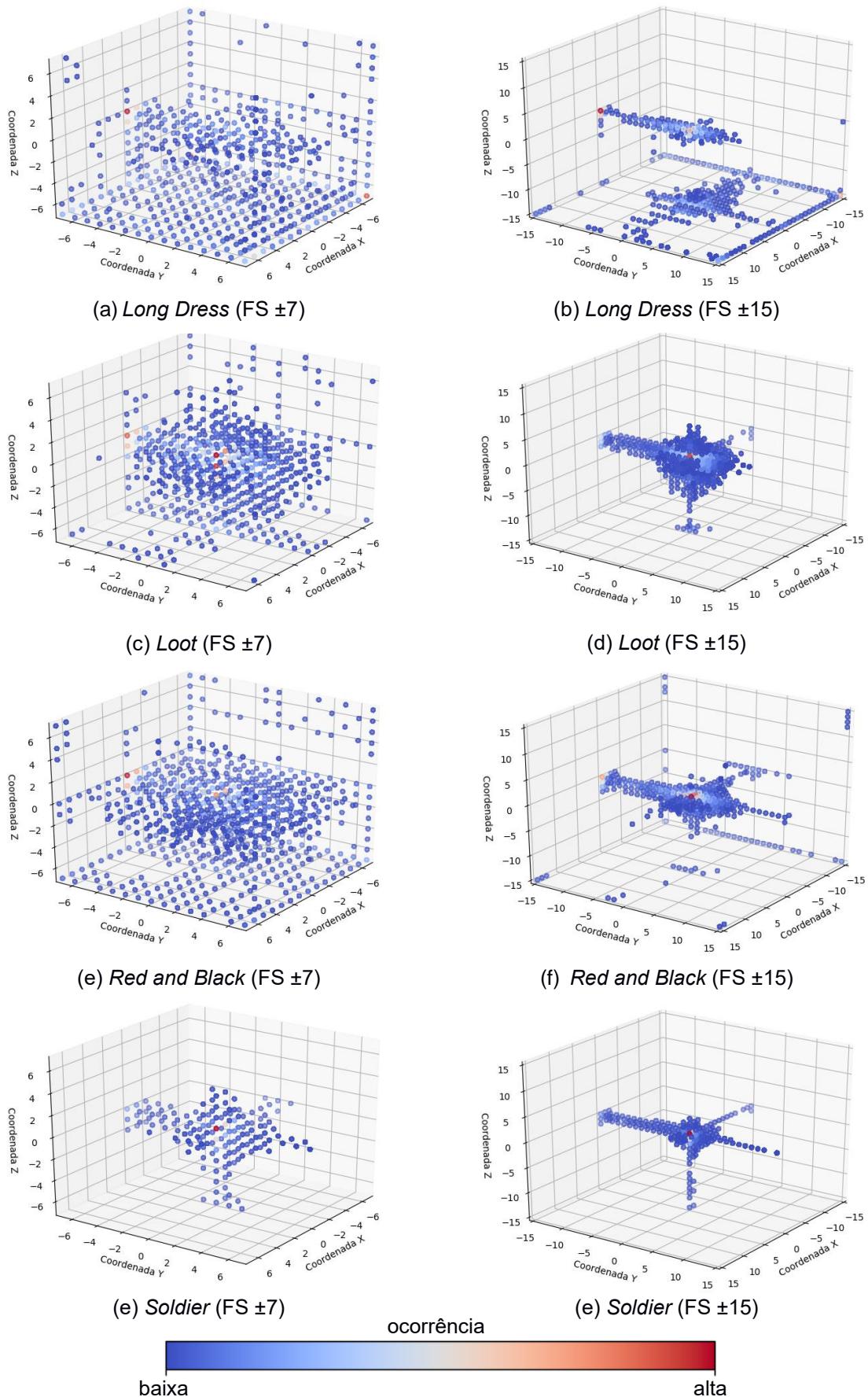


Figura 24: Resultados dos mapas de calor para o algoritmo FS com área de pesquisa  $[-7, +7]$  e  $[-15, +15]$ .

Na Tabela II são apresentados valores de PSNR com P2Plane, assim como P2P para a ME de macroblocos  $32 \times 32 \times 32$  voxels (M32) em comparação com a ME macroblocos  $16 \times 16 \times 16$  voxels (M16), ambos com deslocamento [-15, +15]. Os valores de PSNR P2Plane para FS com M32 em todas as nuvens são inferiores aos valores obtidos por meio do FS com M16 e o mesmo ocorre nos resultados de PSNR P2P.

Tabela II: Avaliação comparativa de PSNR entre o algoritmo FS com blocos de  $32 \times 32 \times 32$  e  $16 \times 16 \times 16$  voxels.

Seqüência	P2Plane (dB)		P2p (dB)	
	M32	M16	M32	M16
<i>Long Dress</i>	68,47	71,64	59,85	61,44
<i>Loot</i>	74,10	76,03	69,49	71,67
<i>Red and Black</i>	69,18	71,61	62,48	69,18
<i>Soldier</i>	75,62	76,44	71,62	72,18
<b>Média</b>	<b>71,84</b>	<b>73,93</b>	<b>65,86</b>	<b>68,62</b>

A Tabela III apresenta resultados de tempo para FS com M32 em comparação ao FS com M16. Os valores de tempo para FS com M32 são superiores aos tempos de codificação apresentados pelo FS com M16, onde o FS com M32 resulta em um aumento médio próximo de 60%. Essa maior demanda de tempo do FS com M32 em relação ao M16 ocorre principalmente, devido a necessidade de maior consumo de tempo para calcular o MSE de blocos candidatos no FS com M32 em relação ao FS com M16. Além disso, em média com FS M32 são avaliados 23356 blocos candidatos por bloco corrente, enquanto que FS com M16 avalia em média 17051 blocos candidatos por bloco corrente. Isso porque blocos de tamanho  $16 \times 16 \times 16$  são mais suscetíveis a contemplar regiões de voxels não ocupados. Outro aspecto importante para o custo computacional elevado do FS com M32 em relação ao FS M16 pode estar relacionado a realização do cálculo do vetor de movimento por meio do algoritmo ICP, que com maior número de pontos tende a desprender maior esforço computacional.

No entanto, embora o FS com M16 apresente melhores resultados em termos de qualidade e menor tempo, o FS com M32 pode habilitar maiores ganhos em compressão, devido a uma maior quantidade de voxels que podem ser representadas por um único vetor de movimento.

Tabela III: Avaliação comparativa de tempo (minutos) do algoritmo FS.

Sequências	FS M32	FS M16
<i>Long Dress</i>	6161	3701
<i>Loot</i>	7922	7441
<i>Red and Black</i>	7212	3542
<i>Soldier</i>	12386	6941
<b>Média</b>	<b>8420</b>	<b>4731</b>

O uso de tamanhos diferentes de blocos, ou seja, a abordagem de subparticionamento proposta neste trabalho pode trazer ganhos em taxa de bits e viabilizar qualidade em blocos menores com maior nível de detalhes. Uma exploração mais detalhada será apresentada no Capítulo 7, comparando resultados de qualidade e taxa de compressão.

Além disso, diante do alto custo computacional apresentado pelo FS, a exploração de algoritmos rápidos de ME é imprescindível para a redução de complexidade, assim como em compressão de vídeo 2D. Neste sentido, o Capítulo 5 apresenta uma avaliação considerando alguns algoritmos rápidos clássicos da codificação de vídeos adaptados para compressão de nuvens de pontos dinâmicas. O objetivo é encontrar o algoritmo mais adequado para redução de complexidade da ME e que apresente a qualidade objetiva mais próxima possível do algoritmo FS.

## 5 AVALIAÇÃO DE ALGORITMOS DE ESTIMAÇÃO DE MOVIMENTO 3D EM NUVENS DE PONTOS

O algoritmo FS fornece resultados de referência para a ME, uma vez que encontra o resultado ideal em uma determinada área de busca. No entanto, apresenta um alto custo computacional devido à pesquisa exaustiva realizada na área de busca (LI, 2004). O alto custo computacional do algoritmo FS é bem conhecido na codificação de vídeo, mas considerando o espaço em 3D das nuvens de pontos, sua complexidade se torna ainda maior. O algoritmo FS com uma busca de [-15, +15] (para cada um dos três eixos) e trabalhando com blocos  $32 \times 32 \times 32$ , pode exigir a avaliação de até 29.791 blocos candidatos para encontrar a melhor correspondência para um único bloco (esse valor é variável já que as nuvens de pontos podem possuir voxels ocupados e não ocupados). Considerando uma nuvem de pontos dinâmica típica das CTC que possui em média 3700 blocos, com uma taxa de 30 nuvens por segundo, deve ser necessário executar até 3,3 bilhões de avaliações de blocos candidatos por segundo para permitir seu processamento em tempo real.

Portanto, a avaliação de algoritmos rápidos de ME sobre as informações da nuvem de pontos 3D pode ser uma maneira interessante de explorar sua correlação temporal, no entanto, com um custo computacional reduzido. Dessa forma, nesta tese os algoritmos rápidos de ME *N-Step Search* (NSS) e *Diamond Search* (DS) são implementados de forma adaptada para operar sobre dados em 3D, aplicados às nuvens de pontos dinâmicas, e seus resultados de qualidade e custo computacional são comparados aos obtidos pelo algoritmo FS. Esses algoritmos foram selecionados por apresentarem baixa complexidade computacional, mas com resultados de qualidade aceitáveis em comparação ao FS.

## 5.1 Algoritmo *N*-Step Search (NSS)

O *N*-Step Search (NSS) é um algoritmo clássico de ME, proposto inicialmente para a compressão de vídeo 2D. A Figura 25 mostra um exemplo da pesquisa realizada pelo algoritmo NSS, considerando *N* igual a três. Os cubos se referem às posições iniciais de cada um dos blocos candidatos. Esta versão do NSS também é conhecida como algoritmo de pesquisa em três etapas (*Three-Step Search - TSS*) (KOGA, 1981).

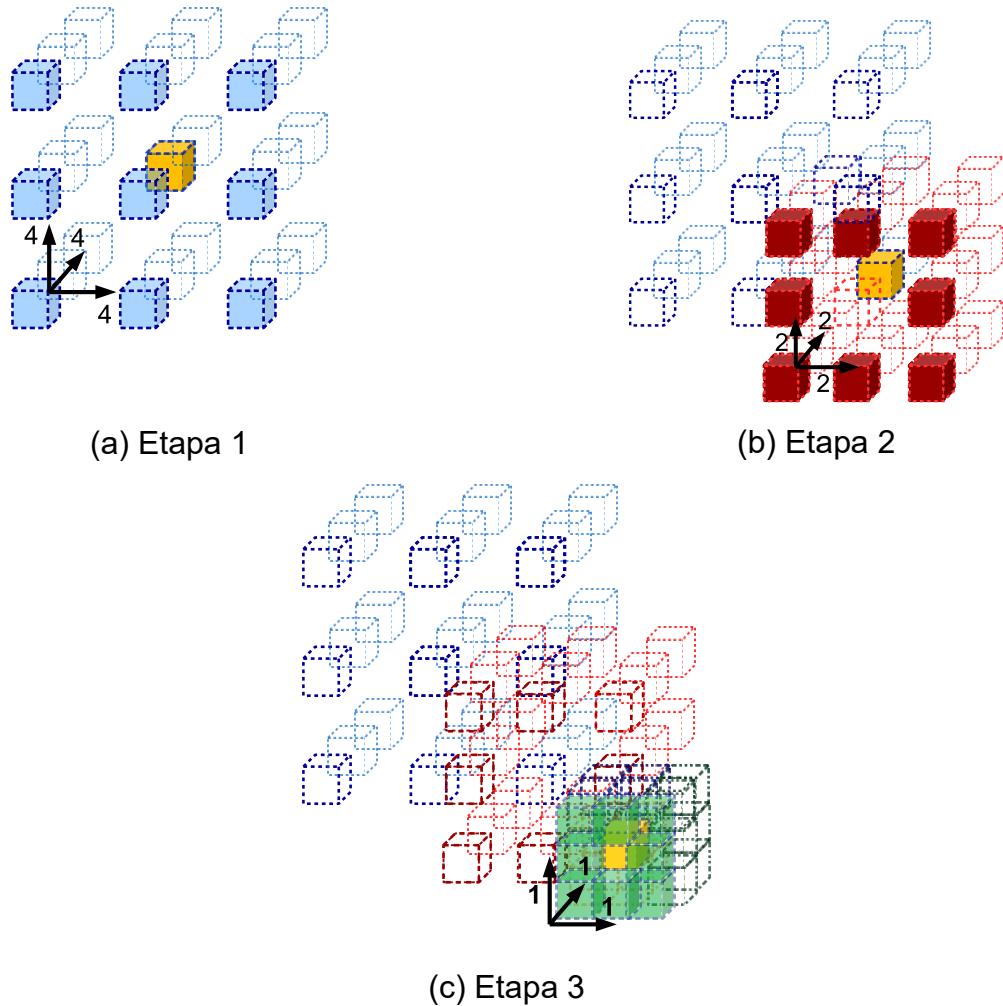


Figura 25: Exemplo de pesquisa com a adaptação do algoritmo TSS para operar no espaço tridimensional.

Na Figura 25 cada bloco se refere a um candidato na área de busca. Na primeira etapa do algoritmo TSS 2D original, nove blocos candidatos devem ser avaliados. No entanto, considerando o espaço 3D (incluindo o eixo z), um total de 27 blocos candidatos devem ser avaliados: a posição central (bloco colocalizado) e os 26 blocos

ao redor do centro, com a distância  $D = 4$ , como mostrado na Figura 25 (a). Na segunda etapa, 26 blocos ao redor do melhor bloco candidato encontrado na primeira etapa devem ser avaliados, desta vez com  $D = 2$ . A Figura 25 (c) apresenta o terceiro passo, que é muito semelhante ao segundo, mas com distância  $D = 1$  entre o centro (melhor resultado no passo dois) e os 26 novos blocos candidatos.

No algoritmo NSS, considerando  $N = 4$ , para melhor cobrir a área de busca de deslocamentos  $[-15, +15]$ , um total 105 blocos candidatos são avaliados, sendo que no FS seriam necessários 29.791 blocos candidatos. Além disso, esse número de blocos candidatos do algoritmo NSS terá um valor máximo dependente de  $N$ .

O NSS pode gerar resultados de qualidade próximos ao FS, porém, esse é propenso a cair em mínimos locais. Mínimo local é um bloco candidato que é selecionado como melhor correspondência (em uma dada região), no entanto não apresenta a melhor correspondência global, que ainda pode possuir melhor correspondência em uma outra região.

## 5.2 Algoritmo *Diamond Search* (DS)

O algoritmo *Diamond Search* (DS) é outro algoritmo rápido clássico de ME (ZHU, 2000). O algoritmo DS emprega dois padrões de pesquisa chamados *Large Diamond Search Pattern* (LDSP) e *Small Diamond Search Pattern* (SDSP). A adaptação do LDSP para operar sobre nuvens de pontos 3D é apresentada na Figura 26 (a), que mostra 26 blocos candidatos ao redor do bloco central (colocalizado), destacado em amarelo. Na versão 3D do LDSP foi adotado deslocamento de um ponto do bloco central para os blocos candidatos diagonais entre os eixos X e Y e dois pontos para os demais blocos candidatos. O SDSP avalia 14 blocos candidatos ao redor do centro, todos com deslocamento de um ponto do bloco central.

O LDSP é aplicado ao bloco colocalizado na área de pesquisa, avaliando seus 27 blocos candidatos, conforme apresentado na Figura 26 (a). O padrão é repetido até que o melhor resultado seja encontrado no centro do padrão. Nesse caso, o padrão SDSP é aplicado como uma etapa final de refinamento, avaliando 14 blocos candidatos adicionais, conforme ilustrado na Figura 26 (b). Caso a melhor correspondência durante a aplicação do padrão LDSP seja encontrada em um vértice, 15 novos blocos candidatos serão avaliados, formando um novo LDSP em torno do melhor

resultado da última etapa, conforme apresentado na Figura 26 (c). Quando o melhor resultado é encontrado em uma aresta do LDSP, nove novos blocos candidatos adicionais são avaliados, conforme ilustrado na Figura 26 (d), formando novamente um novo LSDP. É impossível determinar o número total de blocos candidatos avaliados no algoritmo DS, pois isto dependerá do número de iterações do LSDP e do tipo de pesquisa (vértice ou borda) de cada iteração LSDP. No melhor cenário, apenas 41 blocos candidatos devem ser avaliados (27 comparações para LDSP e 14 para SDSP).

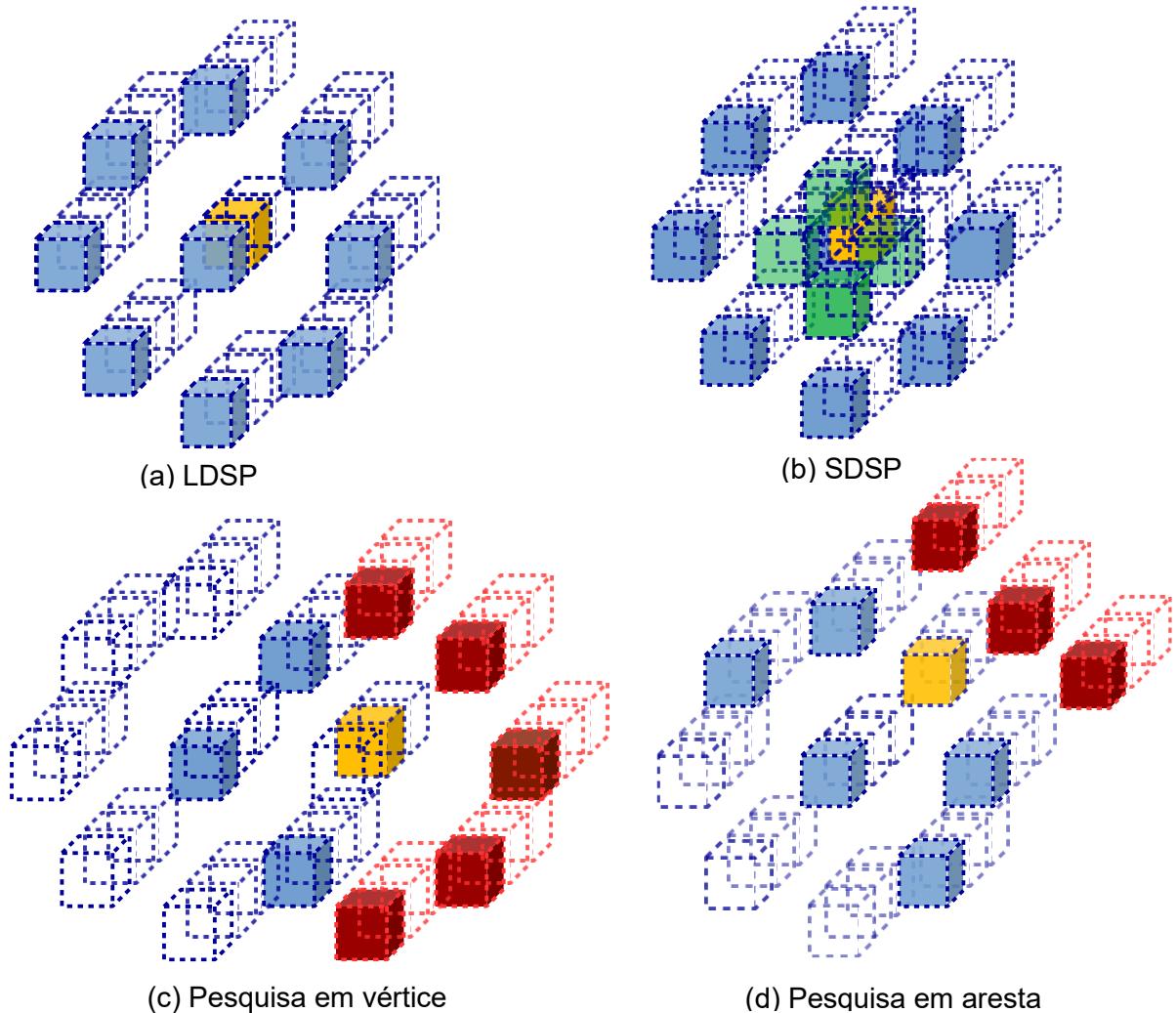


Figura 26: Adaptações do DS para nuvens de pontos.

### 5.3 Avaliação Comparativa

Os algoritmos de FS, NSS e DS foram implementados na linguagem C++ e avaliados considerando o custo computacional e a qualidade de geometria quando

aplicados à nuvem de pontos dinâmicas. A avaliação usa a implementação do *PCC-âncora* como base. Todos os algoritmos de ME executam o ICP, conforme descrito na Seção 3.1, para cada bloco candidato avaliado. Os blocos candidatos são avaliados por meio do MSE simétrico P2P, conforme apresentado no Capítulo 2. Assim, o bloco que possuir o menor MSE resultante do cálculo simétrico é escolhido como melhor correspondência. A abordagem P2P foi escolhida para avaliar os blocos por ser mais facilmente aplicada à avaliação de blocos, já que a abordagem P2Plane exige a re-realização do cálculo dos vetores normais das nuvens de pontos, o que demanda alto esforço computacional. Para essa avaliação foram considerados macroblocos de tamanho  $32 \times 32 \times 32$ , com área de busca com deslocamento de [-15, +15] *voxels* em cada eixo (X, Y, Z), gerando uma área de busca de  $62 \times 62 \times 62$  *voxels*.

Para melhor cobrir o intervalo de pesquisa [-15, +15], nesta tese é considerada a pesquisa NSS, com N igual a quatro. Assim, é denominada pesquisa em quatro etapas (*Four-Step Search - 4SS*). As simulações têm inicialmente  $D = 8$ , sendo que o  $D$  é dividido por 2 até o quarto passo ( $D = 1$ ). Nesse caso, até 105 blocos candidatos podem ser comparados ao bloco atual em cada área de busca.

Os testes foram realizados para as 30 primeiras nuvens das quatro sequências de nuvens de pontos (*Long Dress, Loot, Red and Black e Soldier*) recomendadas pelas *Common Test Conditions* (CTC) (SCHWARZ; CHOU; SINHAROY, 2018). Os resultados de qualidade para cada algoritmo foram medidos em termos de PSNR P2Plane e P2P (SCHWARZ; CHOU; SINHAROY, 2018), enquanto o custo computacional foi medido como o número médio de blocos candidatos avaliados, número médio de iterações de ICP e o tempo total de codificação. Os experimentos foram realizados em um servidor com processador Intel Xeon E5-2650 de 2,20 GHz e 48 GB de RAM.

Na Tabela IV, é possível ver que os melhores resultados de PSNR para as abordagens P2Plane e P2P foram alcançados pelo algoritmo FS, conforme o esperado. O algoritmo FS pode alcançar resultados médios de PSNR P2Plane próximo a 1,8dB e até 2,5dB para PNSR P2P maiores que os algoritmos rápidos. É importante destacar que, o resultado do P2Plane está mais correlacionado com a qualidade subjetiva (JAVAHERI, et al., 2017), por isso o PSNR P2Plane deve ser considerado o principal resultado no aspecto da qualidade.

Tabela IV: Comparação de qualidade (PSNR) entre algoritmos de ME avaliados

Sequências	P2Plane (PSNR (dB))			P2P (PSNR (dB))		
	FS	4SS	DS	FS	4SS	DS
<i>Long Dress</i>	68,47	65,82	65,34	59,85	54,60	56,11
<i>Loot</i>	74,10	73,51	72,85	69,49	66,64	68,75
<i>Red and Back</i>	69,18	66,53	66,58	62,48	57,29	59,02
<i>Soldier</i>	75,62	73,93	75,61	71,62	69,34	69,34
<b>Média</b>	<b>71,84</b>	<b>69,95</b>	<b>70,10</b>	<b>65,86</b>	<b>61,97</b>	<b>63,31</b>

A Tabela V mostra os resultados de custo computacional para os algoritmos de ME avaliados. A coluna “Blocos” e a coluna “ICP” apresenta o número médio de iterações para cada bloco candidato avaliado. Já a coluna “Tempo” total apresenta o tempo total, em minutos, para processar cada nuvem de pontos.

Embora os algoritmos FS e 4SS sejam determinísticos em relação ao número de blocos candidatos avaliados, o número médio de candidatos realmente avaliados durante os experimentos é significativamente menor. Isso ocorre por duas razões principais: 1) os blocos atuais das bordas têm um número limitado de blocos candidatos a serem avaliados, uma vez que a área de pesquisa é limitada na borda e 2) devido aos *voxels* não ocupados, pode ocorrer a inexistência de alguns blocos candidatos, logo, algumas posições de blocos candidatos podem não ser comparados. Isso é comprovado por meio da Tabela V, na coluna “Nº Blocos Comparados”, que apresenta o número médio de blocos candidatos avaliados por cada bloco a ser codificado da nuvem atual. Sendo que, o número médio de candidatos avaliados por bloco no algoritmo FS considerando a média de todas sequências é de 23.356 blocos, enquanto para 4SS e DS o valor médio é de 53,81 e 63,09 respectivamente. Esses resultados mostram que em média são avaliados em torno de 78% dos blocos candidatos no caso do FS e 48,7% no caso 4SS. Além disso, por meio desses resultados é possível verificar que os algoritmos rápidos de ME atingem uma redução de 97,7% de blocos candidatos avaliados em relação ao FS.

Na Tabela V a coluna “ICP” mostra os valores referentes ao número médio de iterações do ICP por bloco candidato avaliado, na média para todas as sequências o FS resulta em 18,46, o que representa o maior número de iterações em comparação ao 4SS e o DS. Esse último possui o menor número de iterações, sendo em média 10,78 iterações por bloco candidato avaliado. Os resultados do número de iterações

do ICP (coluna ICP) juntamente com o número média de blocos avaliados (coluna Nro Bloco), impactam diretamente no resultado de tempo de codificação.

O menor número médio de iterações do ICP por bloco candidato obtido através do algoritmo DS pode ser resultado da configuração adotada para o padrão de busca (formato diamante). A configuração adotada avalia blocos candidatos mais próximos ao bloco central, na comparação com o 4SS, onde no DS o deslocamento dos blocos candidatos que compõem as diagonais nos eixos X e Y possuem deslocamentos de um ponto em relação a posição central, e os demais candidatos apresentam um deslocamento de dois pontos. Já no 4SS, para a primeira etapa, são avaliados blocos candidatos com deslocamentos de oito pontos em cada eixo (X, Y e Z). Assim, a distância entre os blocos candidatos usado no DS pode ter viabilizado um menor número de iterações do ICP e, consequentemente, impactaram em um menor tempo de execução.

Os experimentos mostram que os algoritmos 4SS e DS podem reduzir, em média, o tempo total de ME em 111 e 117 vezes, respectivamente. O algoritmo FS precisa, em média, de 8.420 minutos para processar uma única nuvem de pontos, enquanto os algoritmos rápidos podem fazê-lo em cerca de 74 minutos. Nesse sentido, os algoritmos 4SS e DS apresentam um enorme impacto na redução da complexidade quando comparados ao algoritmo FS, como também esperado. No entanto, devido ao espaço 3D, essa redução é muito maior do que a obtida nos vídeos 2D, por exemplo.

Tabela V: Comparação de complexidade entre os algoritmos de ME avaliados.

Sequências	Nº Blocos Comparados (media por bloco)			ICP (media por bloco)			Tempo (minutos)		
	FS	4SS	DS	FS	4SS	DS	FS	4SS	DS
Long Dress	21.733	51,69	86,98	20,41	22,67	17,56	6.161	72	89
Loot	24.086	56,70	56,36	17,79	15,12	9,11	7.922	69	59
Red and Black	23.002	52,30	68,80	17,37	17,00	12,39	7.212	66	71
Soldier	24.605	54,54	40,21	18,29	14,04	4,05	12.386	96	70
<b>Média</b>	<b>23.356</b>	<b>53,81</b>	<b>63,09</b>	<b>18,46</b>	<b>17,21</b>	<b>10,78</b>	<b>8.420</b>	<b>76</b>	<b>72</b>

Considerando o tempo proibitivo do algoritmo FS, os algoritmos 4SS e DS tornam-se alternativas interessantes para a ME 3D aplicada a nuvens de pontos. Mesmo com um número maior de blocos candidatos avaliados por bloco atual, o algoritmo DS apresenta um resultado muito semelhante no tempo total de execução, com uma

vantagem marginal no resultado médio de tempo, quando comparado ao algoritmo 4SS. O resultado pode ser justificado porque o DS apresenta um número médio menor de iterações do ICP para cada bloco candidato em comparação com o 4SS, conforme presente na coluna “ICP”. Além disso, o DS também obteve ganhos no PSNR P2Plane e no PSNR P2P na comparação com o algoritmo 4SS.

As Tabela IV e Tabela V mostram que os algoritmos rápidos têm uma boa relação entre qualidade e custo computacional quando aplicados às nuvens de pontos. Devido aos resultados semelhantes para o PSNR P2Plane e o tempo, o 4SS e o DS apresentam-se como boas opções para serem usadas na ME de nuvem de pontos dinâmica. Neste trabalho, será utilizado o DS, devido a melhores resultados presentes tanto na qualidade quanto no tempo de codificação.

#### 5.4 Avaliação do impacto do ICP

Os algoritmos de ME avaliados na Seção 5.3, utilizam o algoritmo ICP para calcular o vetor de movimento, que contempla rotações e translações no espaço 3D. Os vetores de movimento, conforme o codificador PCC-âncora, demandam 3 componentes de 16 bits para translação e 3 componentes de 16 bits para rotação, totalizando assim 12 bytes (MEKURIA; BLOM; CESAR, 2017). No entanto, neste trabalho foram realizados experimentos para explorar a possibilidade de redução desse número de componentes, com o objetivo de reduzir a demanda por taxa de bits. Para isso, foi avaliado o impacto da remoção do algoritmo ICP da ME, passando assim a ser calculado somente movimentos de translação entre os blocos candidatos e bloco atual. Dessa forma, o movimento do bloco candidato é compensado considerando a sua posição em relação ao bloco atual, ou seja, se o bloco candidato estiver localizado a dois pontos de distância no eixo X em relação ao bloco atual, esse bloco candidato é transladado em dois pontos no eixo X em direção ao bloco atual.

Para esse experimento foi utilizado o algoritmo DS, pelo fato desse ter sido selecionado para compor a proposta deste trabalho, diante dos bons resultados apresentados em relação aos 4SS e FS conforme Seção 5.3. Dessa forma, é realizado um comparativo entre o DS com a utilização do ICP e o DS sem a utilização.

A Tabela VI apresenta resultados de custo computacional e, qualidade calculados por meio dos valores de PNSR P2Plane e o PSNR P2P, para as 60 primeiras nuvens de pontos das quatro sequências de nuvens de pontos dinâmicas que

compõem as CTC, sendo elas, *Long Dress*, *Loot*, *Red and Black* e *Soldier*. Além disso, foi utilizada uma área de busca com deslocamentos de [-15, +15] e macroblocos de tamanho 32×32×32 voxels.

Como pode ser visto na Tabela VI, os resultados de qualidade de PSNR P2Plane do DS com a utilização do ICP são superiores ao DS sem ICP. Por outro lado, os resultados apresentados com a utilização de PSNR P2p para o algoritmo DS sem ICP são superiores ao DS com ICP. Porém, como o PSNR P2Plane é correlacionado com a avaliação subjetiva, esse se torna mais relevante para a comparação. Além disso, o DS com ICP tem uma superioridade de qualidade de PSNR P2Plane de 2,11dB na média para as quatro sequências. Embora o PSNR P2P resulte em uma melhor qualidade para o DS sem ICP, essa diferença é de 0,11 dB.

Tabela VI: Comparativo entre o algoritmo DS com ICP e sem ICP.

Sequência	P2Plane		P2p		Tempo <sub>(min)</sub>	
	Com ICP	Sem ICP	Com ICP	Sem ICP	Com ICP	Sem ICP
<i>Long Dress</i>	65,34	62,33	56,11	56,94	89	44
<i>Loot</i>	72,85	71,54	68,75	68,42	60	47
<i>Red and Back</i>	66,58	63,23	59,02	59,35	72	53
<i>Soldier</i>	75,61	74,88	71,63	71,25	71	61
<b>Média</b>	<b>70,10</b>	<b>67,99</b>	<b>63,88</b>	<b>63,99</b>	<b>73</b>	<b>51</b>

Em termos de custo computacional, a Tabela VI coluna “Tempo” confirmou o que era esperado, o DS sem ICP apresenta o menor tempo médio de execução para as sequências em relação ao DS com ICP.

Neste sentido, é possível concluir que o algoritmo ICP é fundamental para a qualidade do DS. Sendo assim, a proposta deste trabalho utiliza o DS como algoritmo de ME juntamente com o ICP para calcular os vetores de movimento.

## 6 ESQUEMA DE PREDIÇÃO DESENVOLVIDO

Este capítulo apresenta o novo esquema de predição proposto que engloba estimação de movimento sobre o espaço tridimensional, subparticionamento de macroblocos e decisão de modo que define a melhor opção de codificação, levando em conta tanto as informações de geometria quanto de textura. Este esquema proposto possui como características principais:

- **Alinhamento das caixas delimitadoras:** O ajuste do alinhamento visa uma correspondência mais eficiente para blocos colocalizados, bem como viabiliza que blocos presentes nas bordas possam ter blocos candidatos colocalizados e sejam melhor avaliados pela ME.
- **Estimação de Movimento 3D:** O método aplica o algoritmo DS sobre informações de geometria para ME no espaço 3D das nuvens de pontos dinâmicas.
- **Subparticionamento de macroblocos:** Para melhorar a eficiência das previsões, tanto intra como inter-nuvens, também é proposto o subparticionamento de macroblocos. O método de subparticionamento permite partir de macroblocos  $32 \times 32 \times 32$  ou de  $16 \times 16 \times 16$  voxels. Esse macroblocos podem ser subdivididos em blocos menores como  $16 \times 16 \times 16$  e/ou de  $8 \times 8 \times 8$  voxels, sendo possível codificar cada bloco com inter ou intra.
- **Decisão de modo:** Propõe-se também um algoritmo de decisão de modo para escolher a melhor opção de codificação baseada na relação entre taxa de bits e qualidade, tanto para as informações de geometria quanto de textura, considerando as diferentes possibilidades dos modos de subparticionamento e tipo de predição para os macroblocos.

O esquema proposto foi desenvolvido em C++ e integrado ao codificador PCC-âncora (MEKURIA; BLOM; CESAR, 2017). A modificação no fluxo original do codificador PCC-âncora é mostrada na Figura 27, com um novo bloco de “ME”, contemplando ME por meio do algoritmo DS, bem como o bloco “Subparticionamento de macroblocos” e a “Decisão de modo”. Além disso, há contribuições de melhorias na etapa de “Alinhamento das caixas delimitadoras”. Esses novos blocos para o esquema proposto encontram-se destacados na cor amarela.

O novo esquema de ME utiliza o algoritmo ICP para calcular o vetor de movimento para cada bloco candidato avaliado pelo algoritmo de ME. Além disso, o vetor de movimento obtido através da ME do maior macrobloco do subparticionamento, é utilizado também para os blocos menores. Dessa forma, os blocos menores não realizam ME, somente herdam o vetor de movimento, por exemplo, do macrobloco  $32 \times 32 \times 32$  conforme ilustra a Figura 27. Nesse sentido, é possível refinar a predição inter através de blocos menores, provendo uma melhor eficiência de codificação em algumas regiões da nuvem, já os blocos menores que compõem a região do macrobloco podem possuir uma melhor correlação, e não necessitam de vetores de movimento adicionais. As etapas do esquema proposto serão descritas em detalhes nas próximas seções desse capítulo.

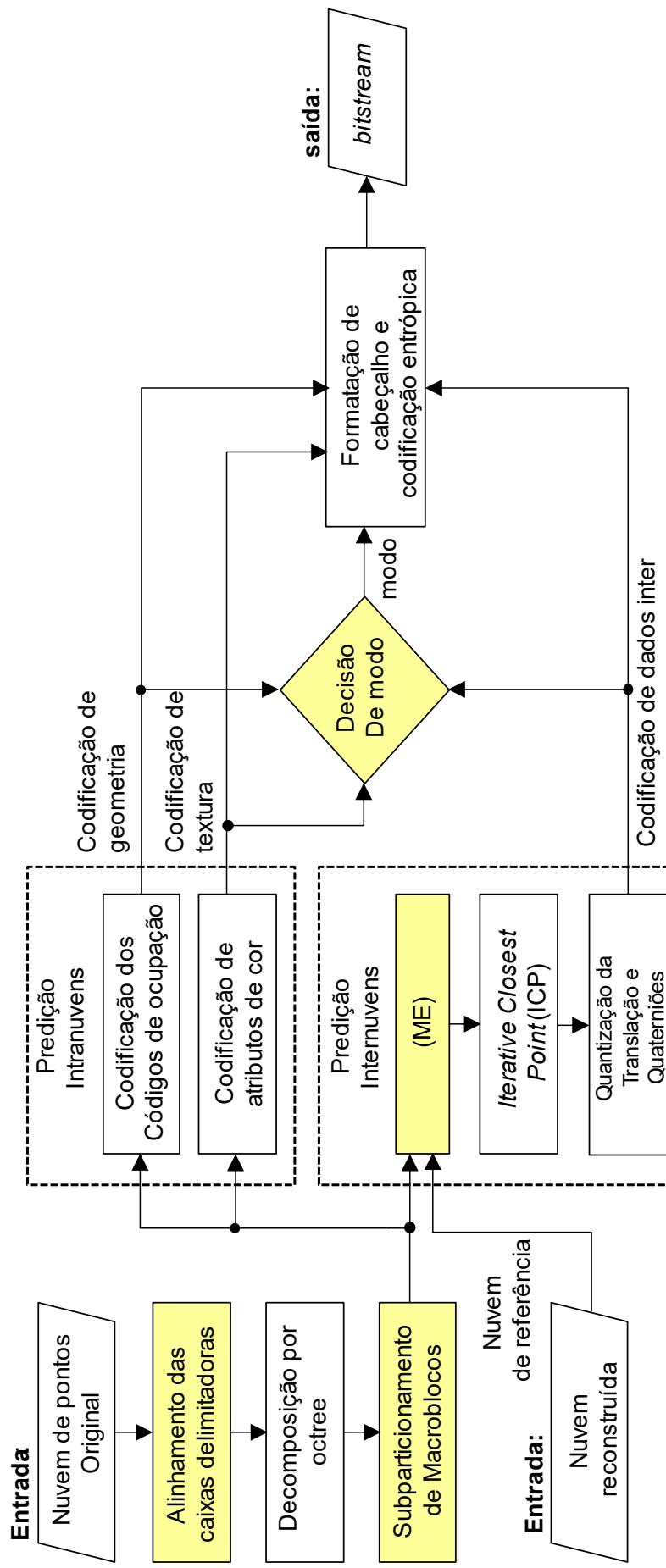


Figura 27: Visão geral do método proposto.

## 6.1 Alinhamento das caixas delimitadoras

No PCC-âncora, a etapa inicial e fundamental para a estimação de movimento é denominada normalização das caixas delimitadoras (*bounding box normalization*). Essa etapa visa alinhar nuvens de pontos temporalmente vizinhas por meio de uma normalização. Para isso, são obtidos os maiores e menores limites em cada eixo (X, Y e Z) considerando duas nuvens de pontos temporalmente vizinhas e esses limites são usados para realizar a normalização.

A normalização é importante principalmente para a abordagem de predição inter-nuvens considerando macroblocos colocalizados, conforme proposta original do PCC-âncora (MEKURIA; BLOM; CESAR, 2017). No entanto, por meio de experimentos realizados previamente a este trabalho, foram detectados problemas de alinhamentos entre nuvens de pontos temporalmente vizinhas. Em especial o problema de alinhamento resulta na ausência de macroblocos colocalizados em regiões próximas às bordas das nuvens causando impacto na eficiência da codificação.

O problema de alinhamento e normalização das caixas delimitadoras ocorre porque nuvens temporalmente vizinhas possuem diferentes limites mínimos e máximos. Essa diferença é devida as nuvens de pontos possuírem estruturas irregulares (*voxels* ocupados e não ocupados), e porque o objeto da cena pode sofrer deformações de nuvem para nuvem. Embora a normalização utilize os maiores e menores limites entre as duas nuvens de pontos temporais, esses limites, após a etapa de normalização, passam a ter valores diferentes, ainda que em casas decimais. Nesse contexto, no processo de geração de macroblocos por meio de *octree*, as coordenadas desses macroblocos podem ser diferentes para nuvens de pontos temporalmente vizinhas.

O problema de alinhamento das caixas delimitadoras do método original do PCC-âncora pode ser visto na Figura 28, onde são ilustradas duas nuvens temporalmente vizinhas sobrepostas. Na Figura 28 as linhas em verde representam os macroblocos de duas nuvens de pontos sobrepostas e através desses macroblocos é possível perceber o desalinhamento. As bordas das caixas delimitadoras das nuvens de pontos sobrepostas estão ilustradas em linhas pretas. Por meio das bordas é possível ver que as caixas delimitadoras estão desalinhadas. Embora os macroblocos de duas

estruturas de *octree* estejam sobrepostas, existe um desalinhamento ou deslocamento entre os macroblocos.

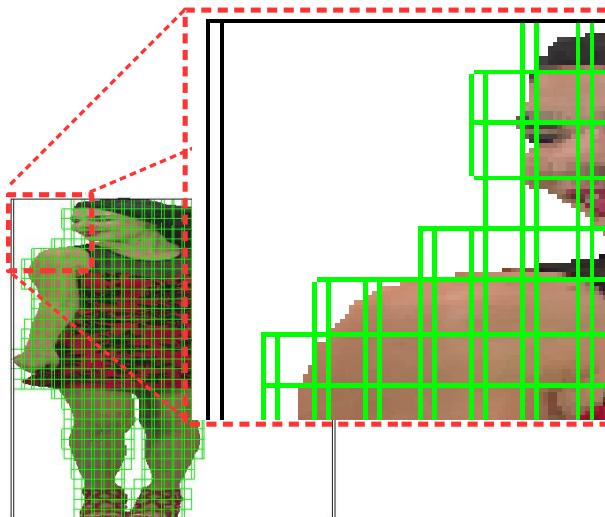


Figura 28: Problema de alinhamento das caixas delimitadoras (adaptado de (Santos, et al., 2018))

Neste sentido, foi proposto fixar os limites das nuvens em coordenadas 0 e 1. Esse ajuste possibilitou uma decomposição por meio de *octree*, como já detalhado na Seção 2.4, respeitando esses limites fixados e forçando a subamostragem dos pontos para o mesmo intervalo, consequentemente as coordenadas dos macroblocos passam a ter as coordenadas dos pontos mais semelhantes.

A Figura 29 mostra o resultado da proposta de correção do alinhamento dos macroblocos, em que é possível ver os blocos perfeitamente sobrepostos e alinhados. Além disso, é possível perceber que as bordas das caixas delimitadoras das nuvens de pontos estão completamente sobrepostas.

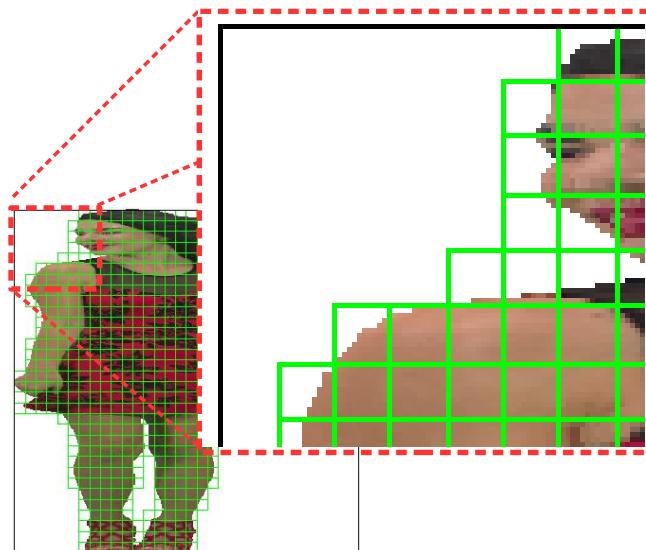


Figura 29: Ajuste das caixas delimitadoras (adaptado de (Santos, et al., 2018)).

Dessa forma, nesta tese é definido o intervalo de limites das caixas delimitadoras entre 0 e 1. Assim, os macroblocos gerados por meio da *octree* proporcionam um melhor alinhamento dos macroblocos. Essa solução viabiliza encontrar blocos colocados em regiões próximas às bordas das nuvens e, consequentemente, habilita maior eficiência de codificação, conforme resultados que serão apresentados no Capítulo 7.

## 6.2 Estimação de Movimento 3D

A correlação temporal de nuvens de pontos dinâmicas é muito maior que a correlação espacial de uma única nuvem de pontos (Li; Li, 2019). Diante disso, a ME é essencial para a eficiência de compressão de nuvens de pontos dinâmicas conforme destacado em (THANOU; CHOU; FROSSARD, 2016).

Neste trabalho foi proposta uma ME que aplica o algoritmo DS forma adaptada para nuvens de pontos conforme descrito Seção 5.2. Esse algoritmo de ME foi selecionado pelo bom desempenho apresentado diante das avaliações descritas na Seção 5.3. Além disso, é proposta a utilização de uma área de busca que contemple deslocamentos [-15, +15] como discutido nos resultados comparativos na Seção 5.3.

A ME 3D proposta utiliza o algoritmo ICP para o cálculo dos vetores de movimento (MV – *Motion Vector*) que, conforme avaliação apresentada na Seção 5.4, é uma abordagem indispensável para que se adquira qualidade de geometria na reconstrução das nuvens de pontos na etapa de decodificação. A métrica de similaridade utilizada no algoritmo DS é o MSE, por meio da abordagem P2P, como descrito na Seção 5.4.

A ME em 3D contempla um bloco novo no fluxo do PCC-âncora, que implementa o algoritmo DS sobre macroblocos de tamanho  $32 \times 32 \times 32$  ou  $16 \times 16 \times 16$ , sendo o tamanho do macrobloco inicial um parâmetro configurável. Na Figura 30 é ilustrado o fluxo da ME proposta. Conforme exposto pelo fluxo, o macrobloco de  $32 \times 32 \times 32$  da nuvem corrente é um dado de entrada que deve passar pelo algoritmo DS. Para cada candidato avaliado no algoritmo DS, primeiramente é calculado o vetor de movimento por meio do algoritmo ICP. Assim, é possível compensar o movimento do macrobloco na nuvem de referência em relação ao macrobloco atual. Posteriormente, para cada macrobloco candidato é calculado o MSE em relação ao macrobloco atual. O

macrobloco que obtiver o menor MSE entre todos os macroblocos candidatos avaliados é selecionado como melhor correspondência. Esse macrobloco selecionado tem o seu vetor de movimento quantizado e juntamente com as suas coordenadas constituí um macrobloco  $32 \times 32 \times 32$  inter (MB-32).

O MB-32 é utilizado pelo subparticionamento que gera macroblocos menores como, por exemplo, macroblocos de  $16 \times 16 \times 16$  e  $8 \times 8 \times 8$ . Cada bloco menor deve ser avaliado por meio de uma decisão de modo para determinar se o bloco será codificado com a predição inter ou intra. Todos esses blocos menores devem constituir um macrobloco de tamanho  $32 \times 32 \times 32$ , onde os blocos subparticionados em  $16 \times 16 \times 16$  irão consolidar um único macrobloco  $32 \times 32 \times 32$  (MBS-16), podendo conter os modos intra e inter, definido pela decisão de modo. Os blocos subparticionados em  $8 \times 8 \times 8$ , inter ou intra, conforme a decisão de modo, irão consolidar um outro macrobloco  $32 \times 32 \times 32$  (MBS-8). O subparticionamento de macroblocos será tratado com maiores detalhes na Seção 6.4.

Na Figura 30 os blocos “DS”, “ICP” e “MSE”, juntamente com a seleção do menor MSE, constituem o fluxo principal da ME 3D proposta neste trabalho. Conforme o fluxo a ME é calculada somente para os macroblocos de tamanho  $32 \times 32 \times 32$ . Os macroblocos MBS-16 e MBS-8 herdam o vetor de movimento do MB-32.

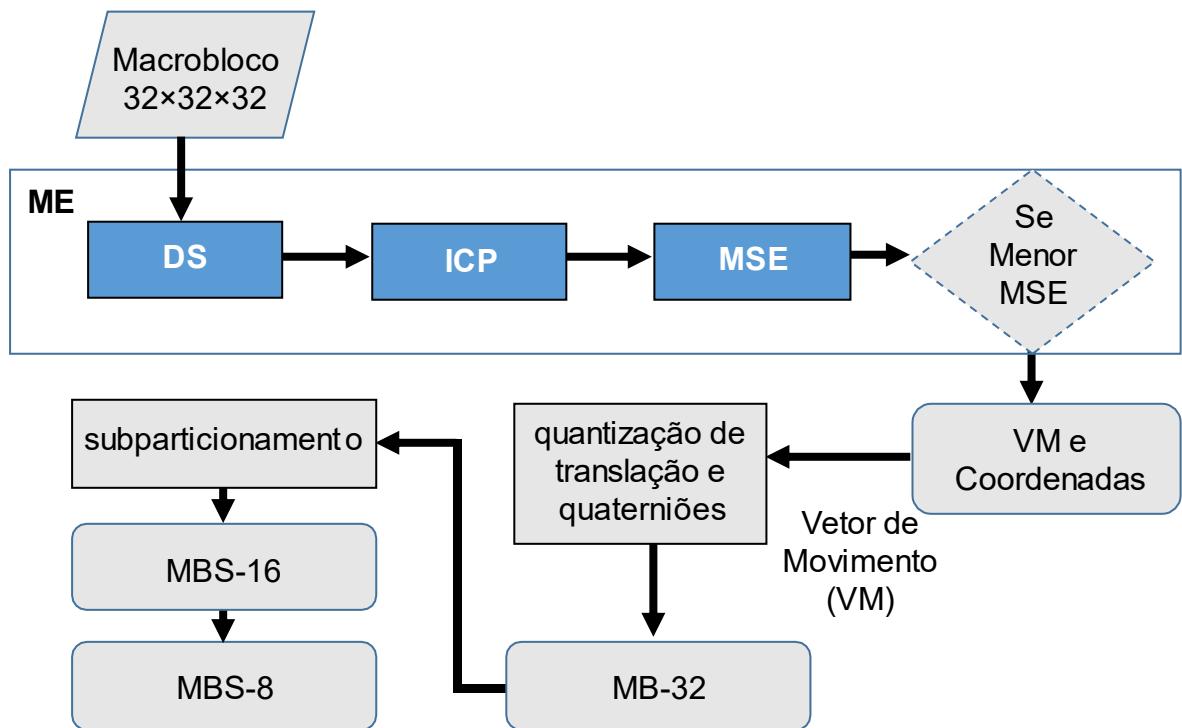


Figura 30: Fluxo da ME 3D.

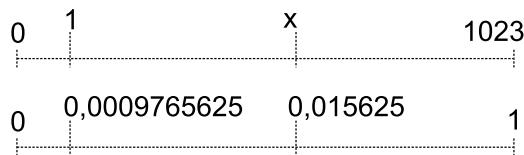
### 6.3 Discretização das coordenadas do macroblocos

No esquema desenvolvido neste trabalho, mais especificamente na ME, as coordenadas dos macroblocos escolhidos como a melhor correspondência são utilizadas juntamente com o vetor de movimento para que possam viabilizar a compensação de movimento na decodificação. A necessidade de transmissão das coordenadas dos macroblocos para o *bitstream* é justificada pela possibilidade de exploração de paralelismo no processo de decodificação, pois dessa forma possibilita a decodificação dos macroblocos sem uma ordem definida. Além disso, as características inerentes às nuvens de pontos, como a irregularidade de *voxels* ocupados, inviabiliza um cálculo diferencial de vetores de movimento entre macroblocos de posições vizinhas, já que nem sempre existem macroblocos vizinhos.

O PCC-âncora apresenta uma etapa de normalização das caixas delimitadoras. Essa normalização tem como objetivo alinhar nuvens de pontos temporalmente vizinhas e permitir uma melhor similaridade em macroblocos colocalizados. Mesmo com a proposta de ajuste de alinhamento discutido na Seção 6.1, ainda é realizada uma normalização que consequentemente resulta em coordenadas em ponto flutuante. No entanto, essas coordenadas demandam alta taxa de bits para sua representação. Assim, neste trabalho é proposta a normalização inversa das coordenadas, resultando assim em coordenadas em intervalos discretos, reduzindo dessa forma a demanda por bits a serem transmitidos.

Na etapa de normalização os intervalos zeros e uns são amostrados para uma determinada resolução, ou seja, *LoD* da *octree*. Na Figura 31 é apresentado um exemplo dessa etapa onde são ilustrados graficamente dois intervalos, o primeiro mais acima representa o intervalo discreto e o intervalo gráfico abaixo referente ao intervalo normalizado (ponto flutuante). Nesse exemplo, é ilustrado como o valor discreto (valor de x) é obtido para a coordenada 0,015625. Com uma resolução geométrica de 10 bits, é obtido como resultado 1024 amostras (intervalos), como ilustra a Figura 31 (linha 1). Considerando 1 como limite do intervalo amostrado, cada amostra tem como resolução o intervalo dividido pelo número de amostras, conforme mostrado na Figura 31 (linha 2). A partir do quociente entre a coordenada de um determinado macrobloco, por exemplo, a coordenada cuja o valor de X é 0,015625, e a resolução das amostras,

nesse exemplo, 0,0009765625, é possível obter 16 como valor discreto da coordenada (Figura 31, linha 3).



1.  $10 \text{ bits} = 2^{10} = 1024 \text{ intervalos}$

2.  $1/1024 = 0,0009765625$

3.  $x = 0,015625 / 0,0009765625 = 16$

Figura 31: Discretização das coordenadas

Nesse sentido, por meio dessa operação aplicada sobre cada um dos eixos que formam a coordenada de um macrobloco (X, Y e Z), é possível transmitir coordenadas discretas para o *bitstream*. Nesse contexto, uma menor quantidade de bits é necessária para representar a nuvem codificada. Além disso, essa abordagem também favorece a representação binária e a exploração de codificação de entropia.

## 6.4 Método de Subparticionamento de Macroblcos

O fluxo original do codificador PCC-âncora permite a codificação de um macrobloco  $16 \times 16 \times 16$ , onde esses podem ser codificados como intra ou inter, tendo em vista alguns critérios como convergência do ICP.

O novo esquema de predição proposto neste trabalho possibilita macroblcos com tamanhos variáveis, ou seja, o subparticionamento de macroblcos. Assim, é possível que o particionamento inicie por macroblcos de tamanho  $16 \times 16 \times 16$  e dessa forma o algoritmo DS é aplicado sobre esses macroblcos, que podem posteriormente ser particionados em blocos de tamanho  $8 \times 8 \times 8$ . Os blocos particionados herdam o vetor de movimento calculado pelo DS por meio do macrobloco inicial de tamanho  $16 \times 16 \times 16$  conforme ilustra a Figura 32.

Também é suportado pelo esquema proposto iniciar o particionamento com macroblcos de  $32 \times 32 \times 32$ , onde através desses será aplicada a ME com DS, os quais posteriormente serão particionados em blocos de  $16 \times 16 \times 16$  e ainda em  $8 \times 8 \times 8$  voxels. Com a utilização de macroblcos de tamanho  $32 \times 32 \times 32$ , por exemplo, o

subparticionamento em  $16 \times 16 \times 16$  voxels é chamado Macrobloco Subparticionado de 16 (MBS-16) e quando particionado em blocos  $8 \times 8 \times 8$  é denominado macrobloco subparticionado de 8 (MBS-8). O MBS-16 e o MBS-8 podem ser comparados diretamente com o macrobloco  $32 \times 32 \times 32$  (MB-32) em termos de taxa de bits e MSE, uma vez que eles têm a mesma quantidade de voxels, ou seja, contemplam a mesma área, como pode ser visto na Figura 33.

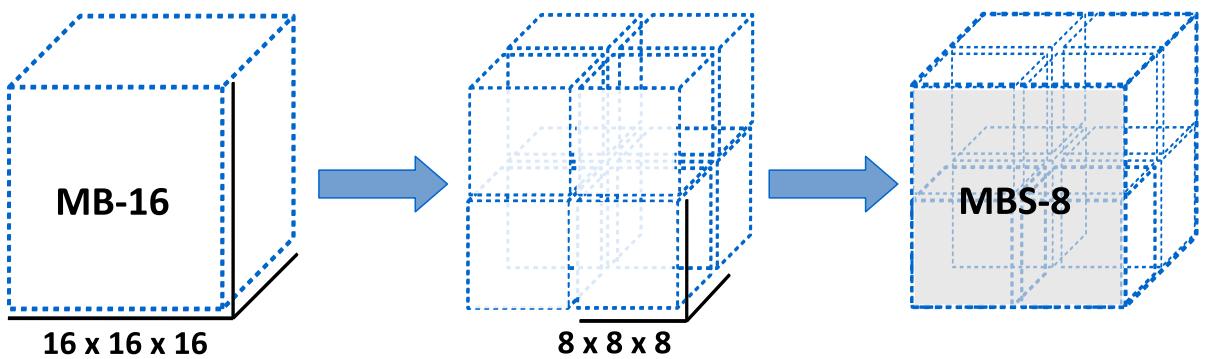


Figura 32: Subparticionamento em 2 níveis.

Os blocos internos do MBS-16, bem como os do MBS-8 podem ser codificados como inter ou intra, sendo assim possível que um macrobloco MBS-16 ou MBS-8 seja composto de blocos intra e inter. Para determinar o melhor particionamento de macroblocos e seu respectivo modo de predição, é necessária uma decisão de modo que atue internamente sobre os blocos. Essa decisão de modo leva em consideração taxa de bits e distorção (MSE) de cada opção.

O método de subparticionamento proposto é capaz de lidar com regiões homogêneas, aplicando macroblocos maiores. Além disso, também pode lidar com regiões mais complexas, como regiões com mais texturas/bordas/movimentos, refinando a predição inter ou intra com o uso de tamanhos de bloco menores, proporcionando maior flexibilidade na predição em comparação ao PCC-âncora.

No exemplo ilustrado na Figura 33, o particionamento inicia com macrobloco de tamanho  $32 \times 32 \times 32$  voxels (MB-32). Para esse tamanho de macrobloco é calculado o vetor de movimento por meio do algoritmo de ME, ou seja, através do algoritmo DS. Posteriormente, esse macrobloco é particionado em blocos de tamanho  $16 \times 16 \times 16$  voxels, que herdam o vetor de movimento calculado para o MB-32. Os blocos de  $16 \times 16 \times 16$  podem ser preditos como inter ou intra, decisão que é tomada pela decisão de modo que considera qual a melhor opção tendo em vista qualidade e taxa,

conforme será apresentado em maiores detalhes na Seção 6.5. Esses blocos devem constituir novamente um macrobloco de  $32 \times 32 \times 32$  voxels (MBS-16), porém com blocos  $16 \times 16 \times 16$  que podem ser inter ou intra. Os blocos de  $16 \times 16 \times 16$  voxels podem ainda serem particionados em blocos de  $8 \times 8 \times 8$ , que assim como os blocos  $16 \times 16 \times 16$  podem ser preditos como intra ou inter. Esses blocos devem constituir um macrobloco de  $32 \times 32 \times 32$  voxels (MBS-8).

Neste trabalho o algoritmo de ME é aplicado somente ao maior macrobloco, sendo possível definir o tamanho deste como,  $32 \times 32 \times 32$  ou  $16 \times 16 \times 16$  voxels, a partir de um arquivo de configuração na codificação. A aplicação do algoritmo de ME somente para o maior macrobloco é justificada pelo fato de viabilizar a menor demanda de bits, já que assim o vetor de movimento (Rotação e Translação obtidos pelo ICP) pode ser utilizado por todos os sub-blocos, demandando menos bits em relação à alternativa de calcular e transmitir o vetor de movimento para cada bloco menor. Além disso, a aplicação da ME em blocos menores tende a impactar em um maior custo computacional. Cabe destacar que para trabalhos é previsto uma exploração de estratégias de ME em blocos menores.

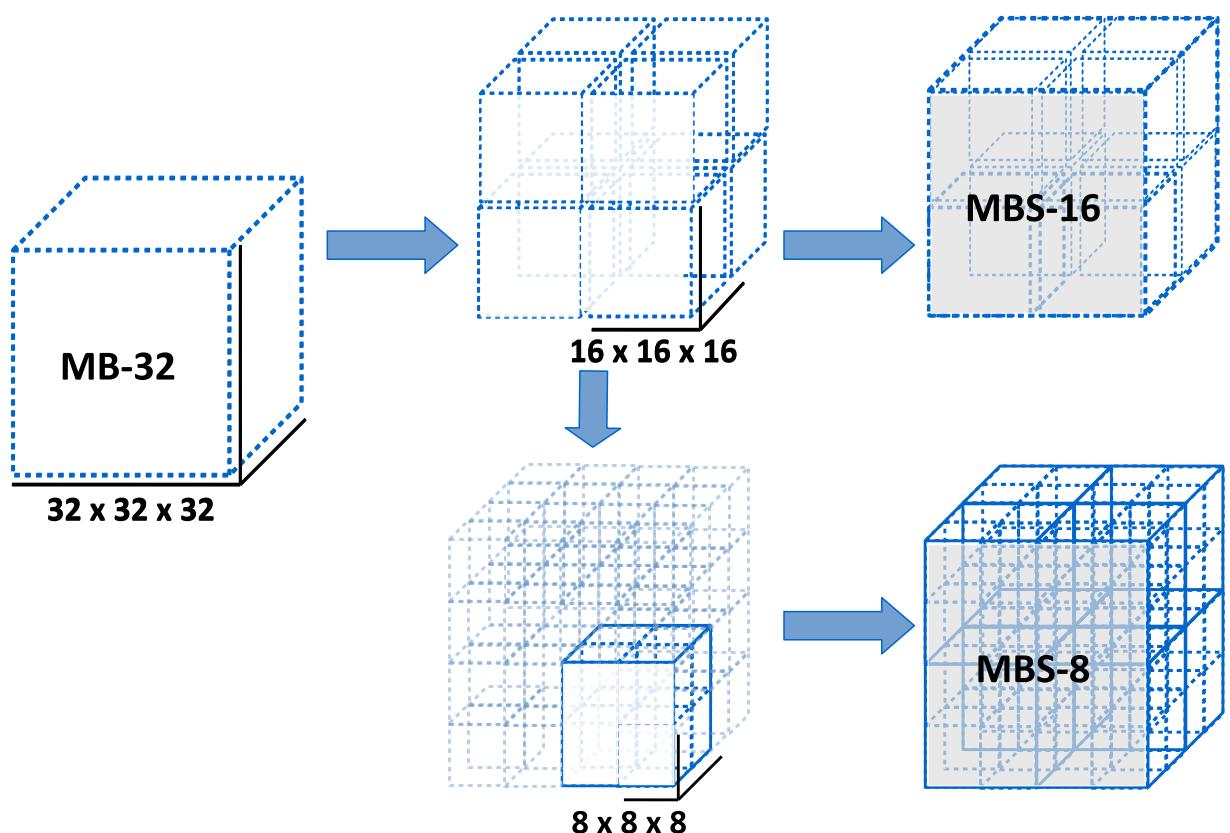


Figura 33: Visão geral da proposta de subparticionamento de macroblocos.

## 6.5 Decisão de Modo

O esquema proposto para compressão de nuvens de pontos dinâmicas oferece muitas possibilidades para codificar um único macrobloco, por exemplo, de tamanho  $32 \times 32 \times 32$  *voxels*. Essas possibilidades incluem o modo de predição inter ou intra e opções de subparticionamento, tornando assim necessário escolher o modo que considera a melhor relação entre taxa de bits e qualidade objetiva. Se apenas a taxa de bits for considerada, o codificador sempre escolherá o modo de predição inter-nuvens para macroblocos  $32 \times 32 \times 32$  *voxels*, pois nesses casos é necessário transmitir somente o vetor de movimento. Além disso, com a predição operando sobre macroblocos maiores existe um número menor de vetores de movimento a serem transmitidos no *bitstream* da nuvem codificada. No entanto, esta opção pode levar a resultados de qualidade insatisfatórios, como ilustrado na Figura 34 (a) (mais resultados subjetivos estão disponíveis no Apêndice B). Na Figura 34 (a) é apresentada uma nuvem de pontos reconstruída por meio de uma codificação que utiliza decisão de modo baseada em taxa de bits do macrobloco inter em relação a taxa de bits do bloco predito como intra (taxa de bits da informação de geometria e textura).

O modo de predição intra pode ser a melhor opção para fornecer melhores resultados de qualidade, uma vez que avaliações realizadas em um trabalho de mestrado do nosso grupo de pesquisa, mas ainda não publicado, mostraram que o valor do MSE tende a ser 10 vezes menor do que o obtido na predição inter-nuvens para *octree* com *LoD* de 10 bits. Porém, macroblocos codificados com a predição intra exigem taxa de bits consideravelmente maiores.

Além disso, o impacto nas informações de textura deve ser considerado, pois uma decisão somente sobre o MSE geometria pode afetar negativamente a textura, conforme ilustra a Figura 34 (b) (resultados subjetivos disponíveis no Apêndice B). Por outro lado, a decisão baseada somente em MSE de textura, pode produzir resultados insatisfatórios para a geometria, como mostrado na Figura 34 (c). Esses resultados são provocados em especial nas regiões de texturas homogêneas, em que a distância entre os pontos vizinhos não afeta o MSE de textura. Porém, a diferença entre o número ou disposição dos pontos pode resultar em regiões de buracos.

Neste sentido, devido às diferentes possibilidades de codificação de um único macrobloco  $32 \times 32 \times 32$  pelo esquema proposto, é necessária uma decisão de modo que permita a codificação eficiente com uma relação entre a taxa de bits e a qualidade

objetiva. A decisão de modo deve escolher a melhor opção de codificação considerando: macrobloco inter de tamanho  $32 \times 32 \times 32$ ; macrobloco intra de tamanho

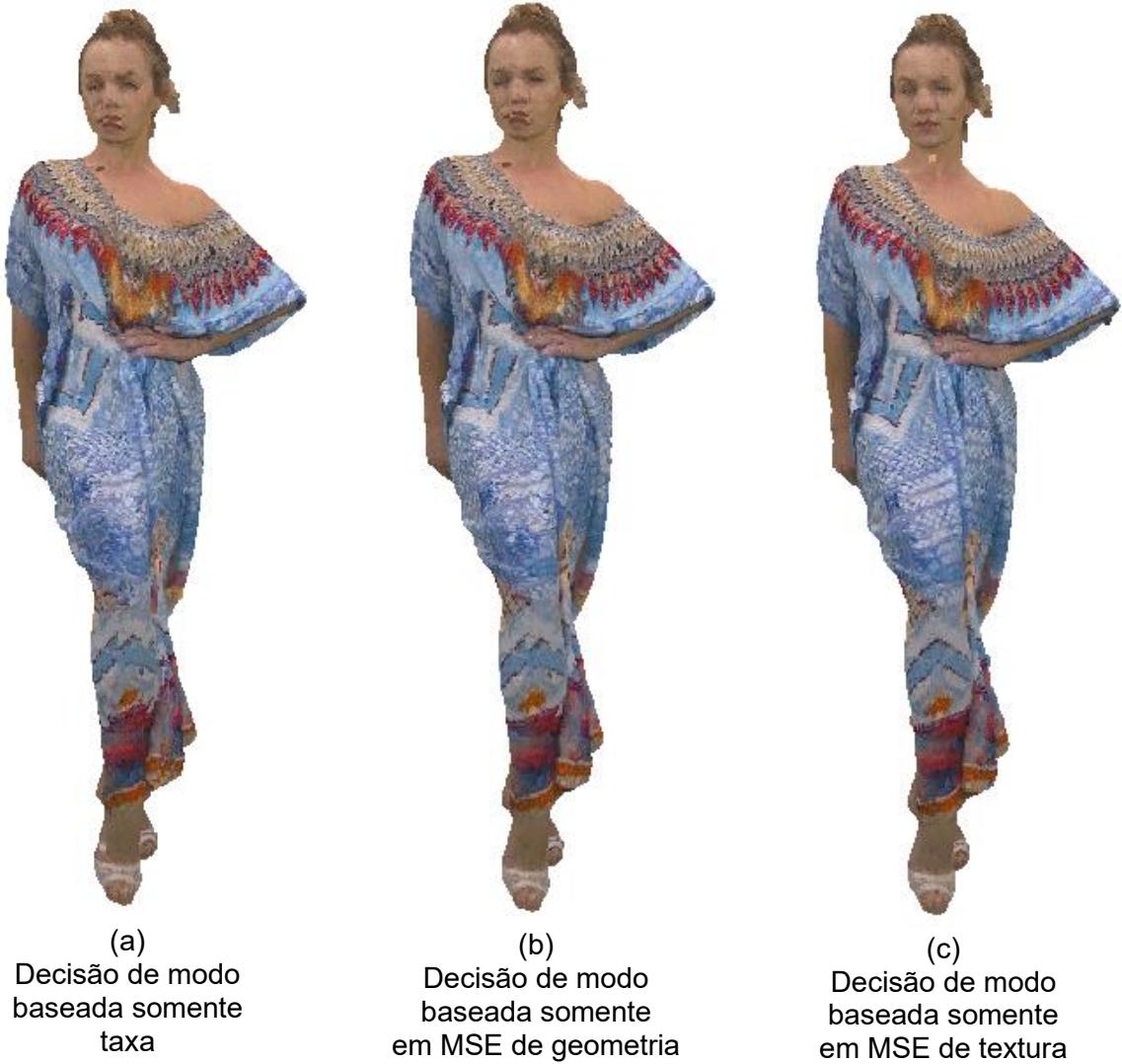


Figura 34: Resultados para diferentes possibilidades de decisão de modo.

$32 \times 32 \times 32$ ; macrobloco MBS-16, incluindo a decisão interna para predição inter ou intra de cada bloco  $16 \times 16 \times 16$ ; ou MBS-8, incluindo a decisão interna para predição inter ou intra de cada bloco  $8 \times 8 \times 8$ . Considerando todos estes aspectos, é apresentado na Figura 35 o fluxo da decisão de modo proposta neste trabalho.

A Figura 35 mostra um exemplo do fluxo da decisão de modo iniciando por um macrobloco de tamanho  $32 \times 32 \times 32$  voxels. Na etapa de ME, o macrobloco  $32 \times 32 \times 32$  é subparticionado em blocos de tamanho  $16 \times 16 \times 16$ . Cada um desses blocos é avaliado considerando taxa e qualidade objetiva para predição intra e também inter. A decisão de modo deve selecionar a melhor opção, e o modo selecionado deve constituir parte do macrobloco MBS-16 (macrobloco  $32 \times 32 \times 32$  formado por blocos  $16 \times 16 \times 16$ ).

O mesmo caminho é percorrido pelos blocos  $8 \times 8 \times 8$  obtido pelo subparticionamento dos blocos  $16 \times 16 \times 16$ . Esses blocos ( $8 \times 8 \times 8$ ) são avaliados considerando taxa e qualidade objetiva para predição intra e inter. A decisão de modo deve selecionar a melhor opção e o modo de predição selecionado é separado para constituir o macrobloco MBS-8. A decisão de modo para os blocos de MBS-16 e de MBS-8 é considerada uma decisão de modo interna, isso porque a decisão de modo externa é responsável por avaliar qual a melhor opção entre os macroblocos de tamanho  $32 \times 32 \times 32$  voxels, ou seja, o MBS-8 (constituído de blocos  $8 \times 8 \times 8$  inter ou intra), MBS-16 (constituído de blocos  $16 \times 16 \times 16$  inter ou intra) e o macrobloco  $32 \times 32 \times 32$  inteiramente intra ou inteiramente predito como inter.

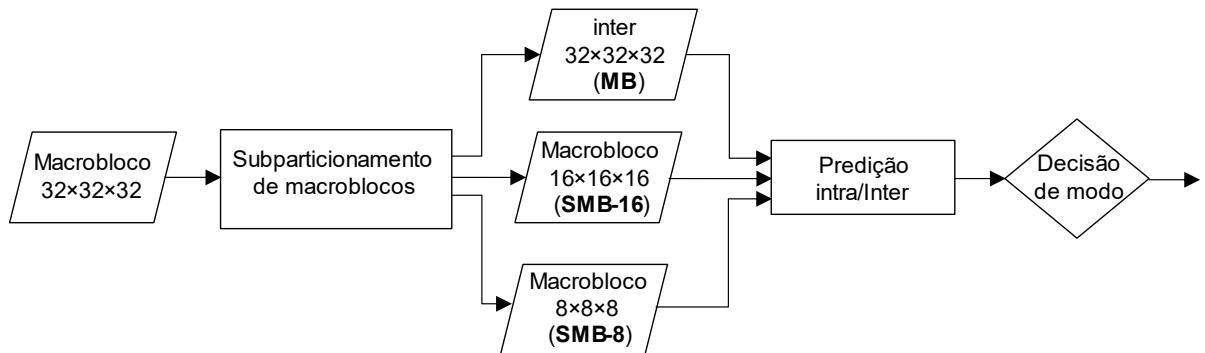


Figura 35: Fluxo da decisão de modo.

Devido à diferença nas faixas de valores de MSE entre a predição intra e inter, foi proposto um fator multiplicador  $\lambda$  para MSE de geometria e textura. A decisão de modo tem um multiplicador  $\lambda$  para MSE de textura intra ( $\lambda_1$ ) e também um multiplicador  $\lambda$  para MSE de geometria intra ( $\lambda_2$ ). Neste contexto, foram propostas funções de custo para qualidade e taxa, para que assim possa constituir uma decisão de modo mais concisa. Para isso, é proposta uma função de custo de qualidade para macroblocos intra  $32 \times 32 \times 32$  como mostrado na Equação 9.

$$MSE_{intra_{32}} = MSE_{tex} \times \lambda_1 + MSE_{geo} \times \lambda_2 \quad (9)$$

Na Equação 9, o  $MSE_{intra_{32}}$  representa a função custo do MSE para macroblocos intra  $32 \times 32 \times 32$ , onde  $\lambda_1$  e  $\lambda_2$  são os fatores multiplicadores de peso para textura e geometria respectivamente.

Com relação aos macroblocos inter 32×32×32, MBS-16 e MBS-8, o cálculo da função custo para MSE é realizado como nas Equações (10), (11) e (12), respectivamente.

$$MSE_{inter_{32}} = MSE_{tex} + MSE_{geo} \quad (10)$$

$$MSE_{SMB-16} = MSE_{tex} + MSE_{geo} \quad (11)$$

$$MSE_{SMB-8} = MSE_{tex} + MSE_{geo} \quad (12)$$

A função de custo de taxa de bits para o macrobloco intra 32x32x32 é calculada com base na estimativa de *bits* por ponto (*bpp*) da nuvem de pontos atual e no número de pontos no macrobloco atual, conforme mostrado na Equação 13.

$$rate_{intra_{32}} = bpp \cdot nro\_ptos \quad (13)$$

A estimativa de *bpp* é obtida por meio da decomposição por *octree* da nuvem de pontos corrente. Na *octree* a quantidade de pontos da nuvem pode ser alterada diante da subamostragem realizada.

A subamostragem da nuvem de pontos corrente é realizada antes da etapa de ME, pois a quantidade de pontos pode influenciar o cálculo do vetor de movimento. Diante disso, a partir da decomposição por *octree* antes da ME torna possível calcular o vetor de movimento para que o bloco possa ser eficientemente compensado na decodificação.

Neste trabalho a estimativa de *bpp* é calculada por meio da aplicação da *octree* na nuvem de pontos corrente para ser utilizada na decisão de modo.

A estimativa de *bpp* é mostrada na Equação (14), onde *total<sub>bits</sub>* é a taxa total em bits estimada para a nuvem de pontos atual e *N<sub>subamostrados</sub>* é o número de pontos da nuvem após a subamostragem.

$$bpp = total_{bits} / N_{subamostrados} \quad (14)$$

Os macroblocos codificados como inter têm um custo fixo de 18 bytes que representam o vetor de movimento e as coordenadas do macrobloco. O vetor de movimento possui três componentes de 16 bits cada para a translação e mais três componentes de 16 bits cada para rotação, conforme apresentado em (MEKURIA; BLOM; CESAR, 2017). As coordenadas do macrobloco, denominadas *Keys*, demandam três componentes de 16 bits.

O custo da taxa de bits para macroblocos MBS-8 e MBS-16 inclui o somatório de todos os vetores de movimento dos blocos da predição-inter constituídos unicamente pelas informações das respectivas *Keys*, representados por ( $MV_{SMB-x}$ ), bem como os dados referentes aos blocos preditos como intra, calculados como na Equação 13. As equações (15) e (16) representam o cálculo do custo da taxa de bits para macroblocos MBS-8 e MBS-16, respectivamente.

$$\text{rate}_{MBS-8} = \sum_{i=0}^n MV_{MBS-8}(i) + bpp \cdot N_{MBS-8} \quad (15)$$

$$\text{rate}_{MBS-16} = \sum_{i=0}^n MV_{MBS-16}(i) + bpp \cdot N_{MBS-16} \quad (16)$$

É importante destacar que os blocos menores inter que constituem os macroblocos, tanto MBS-8 ou MBS-16, herdam o vetor de movimento de rotação e translação do macrobloco de tamanho 32, assim, somente a informação das *Keys* de cada bloco é considerada para calcular o custo por meio de  $MV_{SMB-x}$ .

A função custo da taxa de bits e a função custo de MSE são usados para decidir qual o modo de predição ou opção de subparticionamento é a melhor opção de codificação do macrobloco. Essa avaliação compara uma determinada combinação de modo/subparticionamento na relação do modo de predição intra para o macrobloco 32×32×32, como apresentado no Algoritmo 1.

Em primeiro lugar é confrontado o macrobloco inter 32×32×32 ( $inter_{32}$ ) com o intra também 32×32×32 ( $intra_{32}$ ) (linha 1 do algoritmo 1). Se o  $inter_{32}$  tiver MSE e taxa menor, esse é escolhido para codificação. Por outro lado, é comparado o macrobloco MBS-16 em relação  $intra_{32}$  (linha 3 no algoritmo 1). Se o MBS-16 tiver MSE e taxa menor, será codificado o subparticionamento. Caso contrário, é comparado o MBS-8 com o  $intra_{32}$  (linha 5). Se os modos de predição ( $inter_{32}$ , MBS-16 ou MBS-8) não forem selecionados quando comparados com o  $intra_{32}$ , o macrobloco será codificado como intra (linhas 7 e 8 do algoritmo 1).

---

**Algoritmo 1:** Pseudocódigo de decisão de modo.
 

---

1. **Se** ( $MSE_{inter32} < MSE_{intra32}$  e  $taxa_{inter32} < taxa_{intra32}$ ) **então**
  2.     codificação inter
  3. **Senão Se** ( $MSE_{MBS-16} < MSE_{intra32}$  e  $taxa_{MBS-16} < taxa_{intra32}$ ) **então**
  4.     codificação do subparticionamento MBS-16
  5. **Senão Se** ( $MSE_{MBS-8} < MSE_{intra32}$  e  $taxa_{MBS-8} < taxa_{intra32}$ ) **então**
  6.     codificação do subparticionamento MBS-8
  7. **Senão**
  8.     codificação intra
  9. **Fim Se**
- 

Após concluídas as definições das funções de custo para taxa e qualidade objetiva e o fluxo da decisão de modo, foram realizadas experimentações para definir valores adequados para  $\lambda_1$  e  $\lambda_2$ . Inicialmente o fator multiplicador selecionado teve como base a razão média entre o MSE de geometria para predição inter e intra. Abordagem semelhante também foi aplicada para as informações de textura. Posteriormente para determinar os melhores valores multiplicadores de  $\lambda$  foram realizadas avaliações com diferentes valores de  $\lambda$ , tendo como ponto inicial a razão média do MSE de inter e intra de geometria e também de textura. Para essa experimentação foi usada a sequência de nuvens de pontos *Loot*, onde foi obtido o valor 10 para razão média de geometria e 6 para informações de textura. As experimentações com diferentes variações de valores são apresentadas na Figura 36.

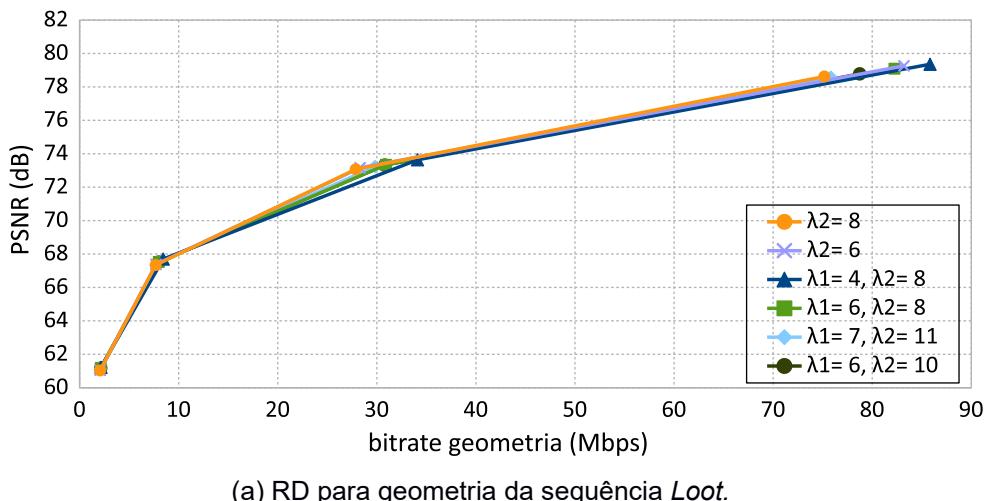
Na Figura 36 são ilustradas as curvas RD da decisão de modo proposta com diferentes valores de  $\lambda_1$  e  $\lambda_2$  (em torno da razão média), cada ponto nas curvas representa o resultado RD de um *LoD* de decomposição de *octree*, sendo os cinco pontos representando g6, g7, g8, g9 e g10, respectivamente (resultados PSNR mais baixos para os mais altos). Nessa avaliação foi utilizado o PSNR P2Plane.

Dado o pequeno número de sequências de testes disponíveis (apenas cinco), foi utilizada apenas uma delas para definir os lambdas, a sequência *Loot*, para evitar viés para as sequências de teste que serão usadas para a avaliação dos resultados finais.

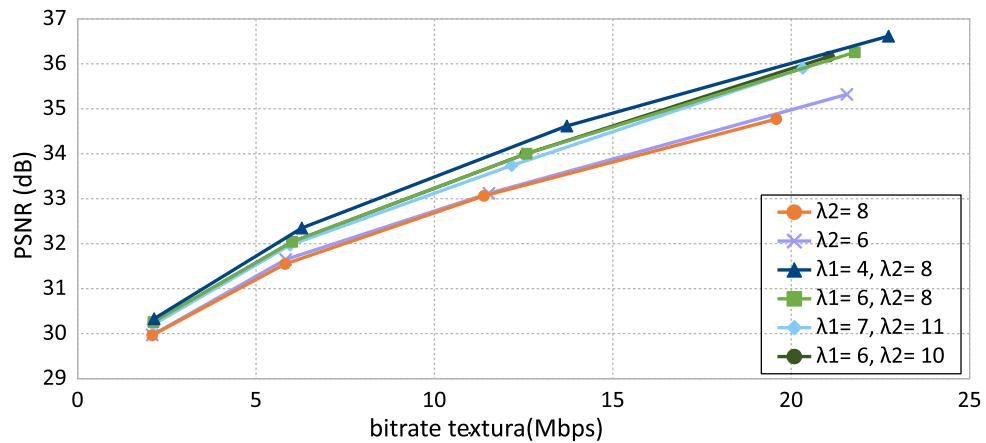
As curvas na Figura 36 (a) mostram que a decisão de modo proposta para valores  $\lambda_2 = 8$  e  $\lambda_2 = 6$  sem multiplicador  $\lambda_1$ , ou seja, a decisão de modo considerando

somente qualidade de geometria obtém melhor qualidade e taxa para os resultados de geometria. Por outro lado, tem o pior resultado para curvas de textura conforme Figura 36 (b). Nesse sentido, é possível perceber a necessidade de usar o MSE de geometria juntamente com o MSE de textura na decisão de modo.

Os resultados das opções que consideram tanto geometria quanto textura para o gráfico RD de geometria Figura 36 (a), apresentam curvas muito similares, podendo ser consideradas pouco conclusivas. Porém, se confrontadas as curvas RD de geometria com as RD de textura, é possível perceber que os resultados para  $\lambda_1 = 4$  e  $\lambda_2 = 8$  apresentam os melhores resultados para o gráfico RD de textura Figura 36 (b), bem como atinge maior qualidade final na curva apresentada pelo RD de geometria. Neste sentido,  $\lambda_1 = 8$  e  $\lambda_2 = 4$  foi selecionado para constituir a decisão de modo utilizada no esquema de predição proposto neste trabalho.



(a) RD para geometria da sequência *Loot*.



(b) RD para textura da sequência *Loot*.

Figura 36: Experimentações para definição do valor de  $\lambda_1$  e  $\lambda_2$ .

## 7 RESULTADOS EXPERIMENTAIS

Neste capítulo serão descritos os resultados obtidos pelo esquema de predição proposto nesta tese. Primeiramente será descrita a base experimental utilizada. Na sequência, serão apresentados os resultados, de forma isolada, para algumas das abordagens propostas e, na seção 7.4, são apresentados os resultados finais, considerando a solução completa proposta.

### 7.1 Base Experimental Utilizada

O esquema de predição proposto foi implementado no software do PCC-âncora e avaliado de acordo com as condições comuns de teste (CTC) (SCHWARZ; CHOU; SINHAROY, 2018). Assim foram usadas as sequências de nuvens de pontos dinâmicas “*Long Dress*”, “*Red and Black*”, “*Loot*” e “*Soldier*” que compõem o *dataset* descrito em (D'EON et al., 2016), e a sequência denominada *Queen*. Segue na Figura 37 a primeira nuvem de pontos de cada uma dessas sequências.

A avaliação foi realizada para todas as nuvens de pontos de cada sequência com cinco *LoD*, variando de 6 (g6) a 10 bits (g10). Os resultados objetivos de qualidade foram gerados usando as abordagens P2Plano e P2P, conforme definido em (SCHWARZ; CHOU; SINHAROY, 2018). Os resultados foram gerados por meio de um servidor Intel XEON E5-2650 de 2,20 GHz com 48 GBytes de RAM.

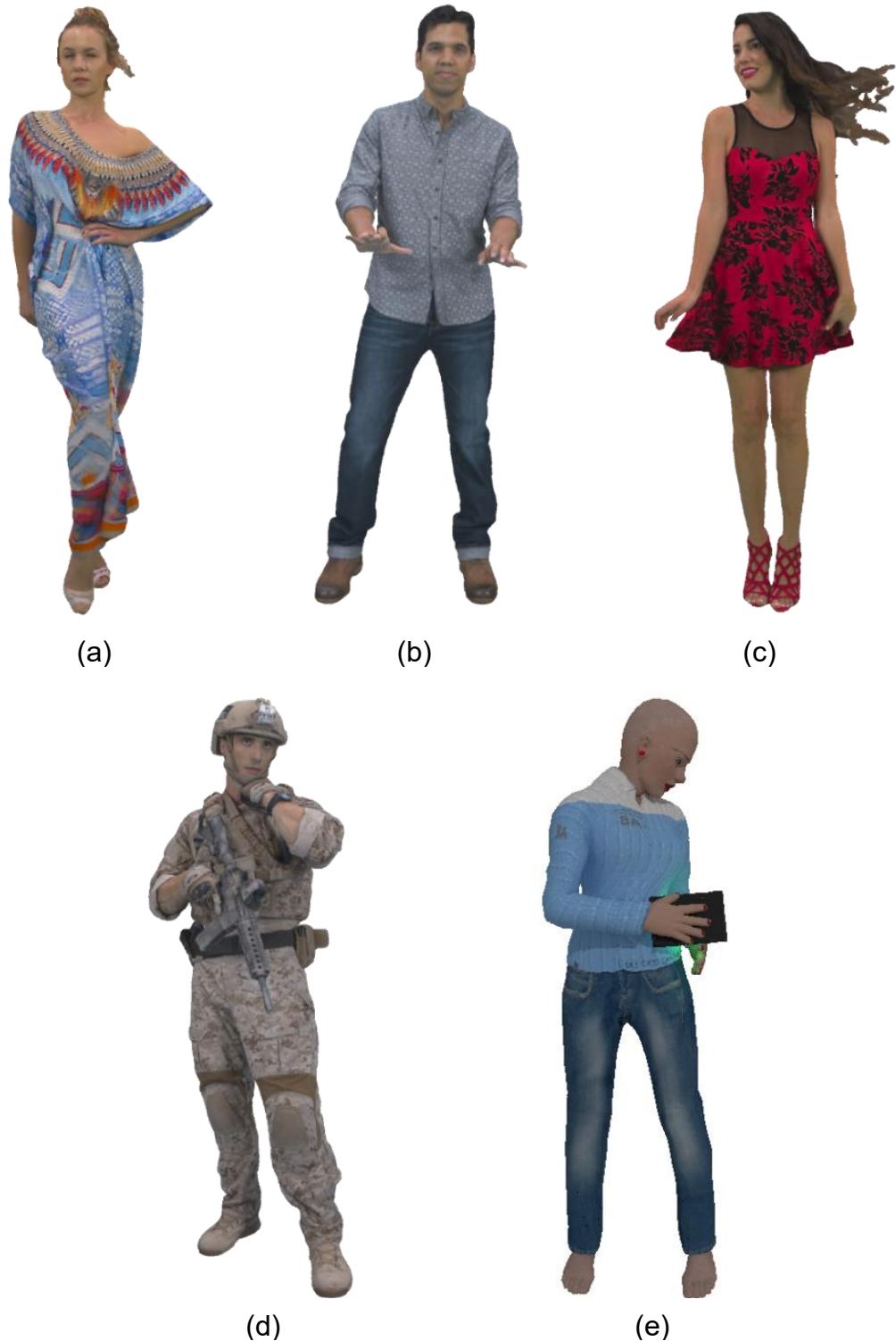


Figura 37: Sequências de nuvens de pontos, *Long Dress* (a), *Loot* (b), *Red and Black* (c), *Soldier* (d) e *Queen* (e).

## 7.2 Resultados do Ajuste das Caixas Delimitadoras

O alinhamento das caixas delimitadoras conforme apresentado na Seção 6.1, viabiliza a ME de blocos colocalizados, principalmente em regiões próximas às bordas das nuvens e, consequentemente, habilita ganhos em compressão. Neste sentido, o ajuste proposto para o alinhamento possibilita aumentar o número de macroblocos colocalizados em relação à proposta do PCC-âncora.

Os resultados deste experimento consideram um comparativo entre o codificador PCC-âncora original e a versão com o alinhamento das caixas delimitadoras proposto neste trabalho. Para coleta desses resultados foram utilizados macroblocos de tamanho  $16 \times 16 \times 16$  para ambas as abordagens. A avaliação de qualidade objetiva de geometria foi realizada por meio do PSNR P2Plane e P2P.

Na Figura 38 é apresentado o gráfico de taxa de bits e qualidade de geometria (*Rate and Distortion* - RD) média considerando as sequências de nuvens de pontos avaliadas. O gráfico possibilita uma visualização mais intuitiva dos resultados das duas propostas de codificação. As curvas do gráfico relacionam taxa e qualidade (PSNR P2Plane) para os diferentes *LoD*, onde os maiores *LoD* apresentam maior qualidade, mas também as maiores taxas. Diante desse gráfico RD é concluir que o alinhamento das caixas delimitadoras apresenta uma maior eficiência de codificação em relação ao PCC-âncora. Os resultados completos desse experimento estão disponíveis no Apêndice A, Figura 55, que apresenta todos os gráficos RD para as sequências avaliadas.

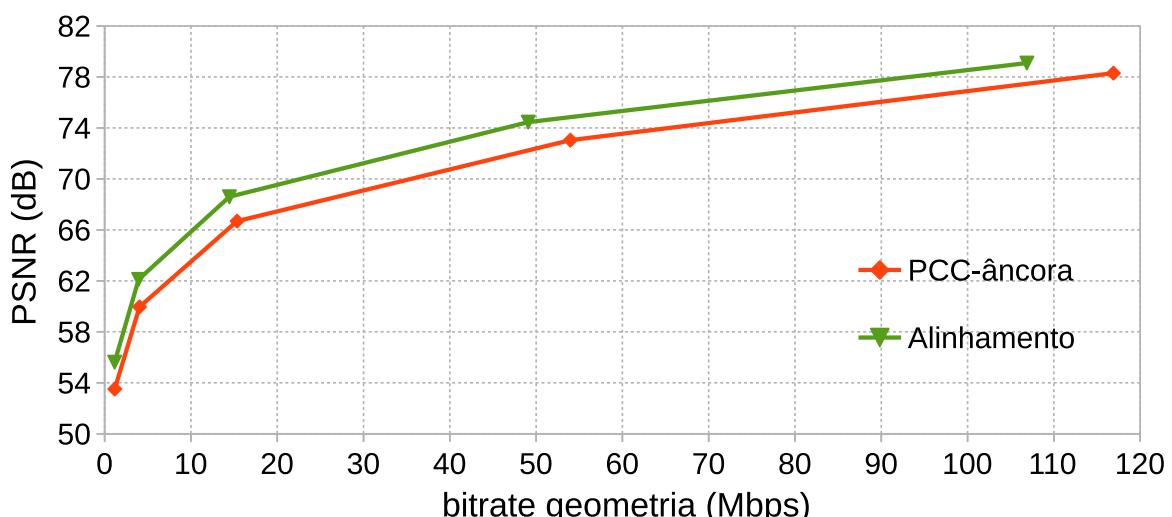


Figura 38: Gráfico RD de geometria para a proposta de alinhamento

Na Figura 39 é apresentado um gráfico RD de textura com resultados médios para as sequências utilizadas neste experimento. Essa figura mostra que de modo geral, como esperado, a proposta de alinhamento apresenta bons resultados tanto em qualidade como em taxa, em comparação com o PCC-âncora.

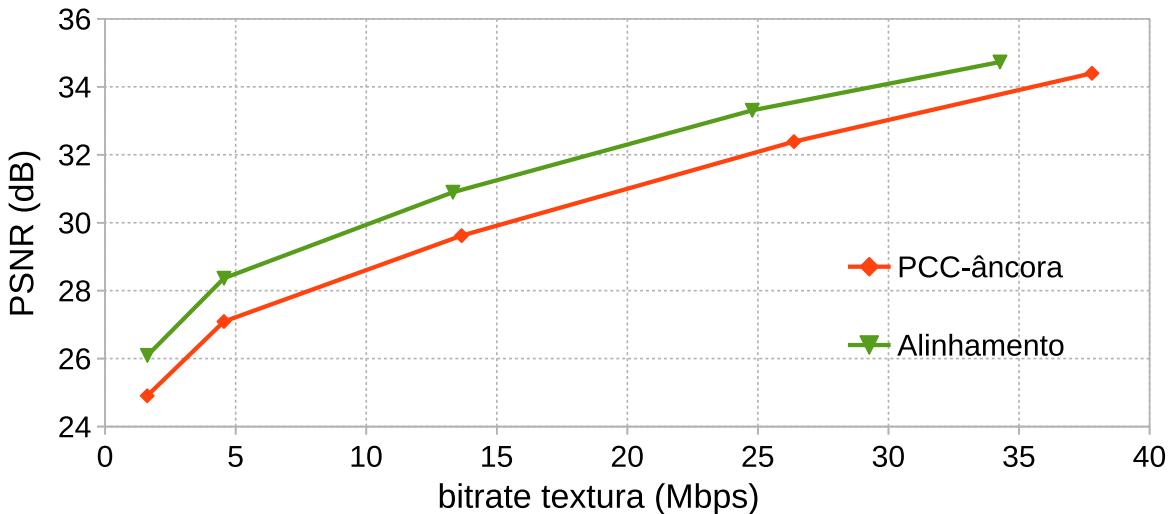


Figura 39: Gráfico RD de textura para a proposta de alinhamento

A Tabela XI apresenta resultados comparativos para a porcentagem do número de pontos preditos como inter para a proposta alinhamento em relação ao PCC-âncora. Os valores são discriminados por sequência de nuvens de pontos, assim como para a média por *LoD*. Optou-se por apresentar o percentual de pontos preditos como inter com o objetivo de viabilizar também a comparação da proposta de alinhamento com outras abordagens, como o subparticionamento de macroblocos que será apresentada na Seção 7.3. Poderia também ser utilizado o percentual de blocos codificados com a predição inter-nuvens. No entanto, o percentual do número de blocos inviabiliza a comparação com a abordagem de subparticionamento, uma vez que os blocos podem ter diferentes tamanhos.

Por meio da Tabela VII é possível concluir que, o alinhamento proposto possibilita o encontro de uma maior quantidade de pontos codificados através da predição inter-nuvens. O resultado da média dos *LoD* mostra que o alinhamento apresenta entre 4% e 19% mais pontos codificados como inter em relação ao PCC-âncora em g6 e g10 respectivamente. O percentual é maior à medida que o *LoD* cresce, isso porque os maiores *LoD* possuem maior quantidade de blocos.

Tabela VII: Percentual médio de pontos codificados por meio da predição inter-nuvens.

Sequências	Abordagens	LoD				
		g6	g7	g8	g9	g10
<i>Long Dress</i>	PCC-âncora	0,00	0,10	1,51	4,21	5,35
	Alinhamento	0,02	0,35	3,77	9,52	13,53
<i>Loot</i>	PCC-âncora	0,98	9,60	13,98	11,75	23,38
	Alinhamento	12,64	29,00	45,96	54,81	52,23
<i>Red and Black</i>	PCC-âncora	1,07	5,34	7,66	13,81	15,79
	Alinhamento	2,02	13,63	26,82	35,05	34,44
<i>Soldier</i>	PCC-âncora	0,00	0,56	2,68	7,15	4,97
	Alinhamento	1,07	1,78	10,16	28,57	53,27
<i>Queen</i>	PCC-âncora	1,82	7,87	15,93	22,83	36,71
	Alinhamento	6,57	22,83	50,33	69,26	78,98
<b>Média</b>	<b>PCC-âncora</b>	<b>0,77</b>	<b>4,70</b>	<b>8,35</b>	<b>11,95</b>	<b>17,24</b>
	<b>Alinhamento</b>	<b>4,42</b>	<b>13,27</b>	<b>25,74</b>	<b>34,52</b>	<b>36,96</b>

Na Tabela VIII são apresentados os resultados obtidos por meio da métrica *Bjontegaard-Delta* (BD) para taxa de bits (BD-Rate) e PSNR (BD-PSNR) de geometria e textura. A métrica BD foi calculada para a proposta alinhamento, em relação ao PCC-âncora para cada uma das nuvens de pontos em g7, g8, g9 e g10. Na Tabela VIII o "geo.BD-PSNR" e o "geo.BD-RATE" referem-se ao BD-PSNR e ao BD-Rate das informações da geometria, considerando as abordagens P2Point (D1) e P2Plane (D2).

A partir dos resultados da Tabela VIII, é possível concluir que o alinhamento proposto pode obter ganhos consideráveis em termos de eficiência de compressão. Os resultados médios de BD-Rate de geometria mostram uma redução de 16,56% e 31,77% para D1 e D2 em relação ao PCC-âncora. Os resultados da geometria BD-PSNR mostram ainda um aumento de 0,87 dB para D1 e 2,04 dB para D2. O "attr.BD-PSNR" e o "attr.BD-RATE" apresentam os valores de BD-PSNR e BD-Rate das informações de textura. Para os atributos de textura, a redução de BD-RATE foi de 29,83%, 9,96% e 12,56% para os componentes Luminância (Luma), Crominância Azul (Cb) e Crominância Vermelha (Cr), respectivamente. Além disso, o BD-PSNR para textura apresentou ganho de 1,29 dB, 0,22 dB e 0,31 dB para os canais Luma, Cb e Cr, respectivamente.

Tabela VIII: Resultados de *Bjontegaard-Delta* para esquema de alinhamento desenvolvido em relação ao PCC-âncora.

Sequência	geo.BD-RATE (%)		geo.BD-PSNR (dB)		attr.BD-RATE (%)			attr.BD-PSNR (dB)		
	D1	D2	D1	D2	Luma	Cb	Cr	Luma	Cb	Cr
<i>Long Dress</i>	-14,57	-33,89	0,80	2,33	-30,26	-4,08	-6,51	1,29	0,06	0,11
<i>Loot</i>	-19,02	-31,29	1,00	1,99	-26,50	-3,57	-5,19	0,95	0,07	0,11
<i>Red and Black</i>	-16,19	-26,11	0,86	1,64	-25,86	-6,13	-12,48	0,99	0,10	0,24
<i>Soldier</i>	-15,35	-33,96	0,79	2,19	-31,22	-1,93	-0,77	1,27	0,02	0,00
<i>Queen</i>	-17,68	-33,59	0,90	2,03	-35,32	-34,07	-37,87	1,95	0,85	1,07
<b>Média</b>	<b>-16,56</b>	<b>-31,77</b>	<b>0,87</b>	<b>2,04</b>	<b>-29,83</b>	<b>-9,96</b>	<b>-12,56</b>	<b>1,29</b>	<b>0,22</b>	<b>0,31</b>

### 7.3 Resultados do Subparticionamento de Macroblocos

O subparticionamento de macroblocos proposto tem como objetivo viabilizar maior flexibilidade nas predições intra ou inter de macroblocos. Essa abordagem é capaz de lidar com regiões homogêneas, aplicando macroblocos maiores. Além disso, também pode lidar com regiões mais complexas, como regiões com mais texturas ou bordas, refinando a predição inter ou intra com o uso de tamanhos de blocos menores.

Os resultados do experimento para o subparticionamento de macroblocos apresentado nessa seção foram realizados para duas configurações: 1) subparticionamento partindo de macroblocos de tamanho  $32 \times 32 \times 32$ , que podem ser particionados em blocos de  $16 \times 16 \times 16$  e que podem ainda ser particionados em blocos de  $8 \times 8 \times 8$  voxels, e 2) subparticionamento iniciando por macroblocos de tamanho  $16 \times 16 \times 16$  voxels, que podem ainda ser particionados em blocos de  $8 \times 8 \times 8$ . Os resultados do esquema de subparticionamento avaliado foi implementado tendo como base a versão do PCC-âncora com o esquema de alinhamento das caixas delimitadoras proposto neste trabalho. Além disso, foi desenvolvida uma decisão de modo para que o codificador possa escolher qual das opções de tamanho de bloco e qual modo de predição será utilizado. Como este experimento tinha como objetivo mostrar os resultados do subparticionamento para o codificador, optou-se por uma decisão de modo baseada somente em taxa de bits. No entanto, os demais critérios de decisão de modo do PCC-âncora ainda foram preservados, conforme apresentado na Seção 3.1. Porém, experimentos preliminares mostram que o critério de variância de cor presente no PCC-âncora excluía praticamente todos os macroblocos inter de tamanho  $32 \times 32 \times 32$ , assim foi necessária a remoção desse critério. Diante da avaliação dos critérios na decisão

de modo, originalmente presentes no PCC-âncora, como similaridade do número de pontos em os blocos e o *fitness score*, é esperado que os blocos sejam codificados com uma das opções de subparticionamento.

Na Figura 40 é ilustrado o gráficos RD com valores médios de geometria para as cinco sequências de nuvens de pontos utilizadas nesse experimento (resultados completos estão disponíveis no Apêndice B). No gráfico RD é possível concluir que o Subparticionamento iniciando com macroblocos de tamanho 32 (Subparticionamento 32) alcança as maiores reduções em taxa de bits em relação ao PCC-âncora e também em relação a opção de subparticionamento iniciando por macroblocos de tamanho 16 (Subparticionamento 16). No entanto, o Subparticionamento 16 apresenta resultados mais equilibrados entre taxa e qualidade mostrando melhores resultados em relação ao PCC-âncora.

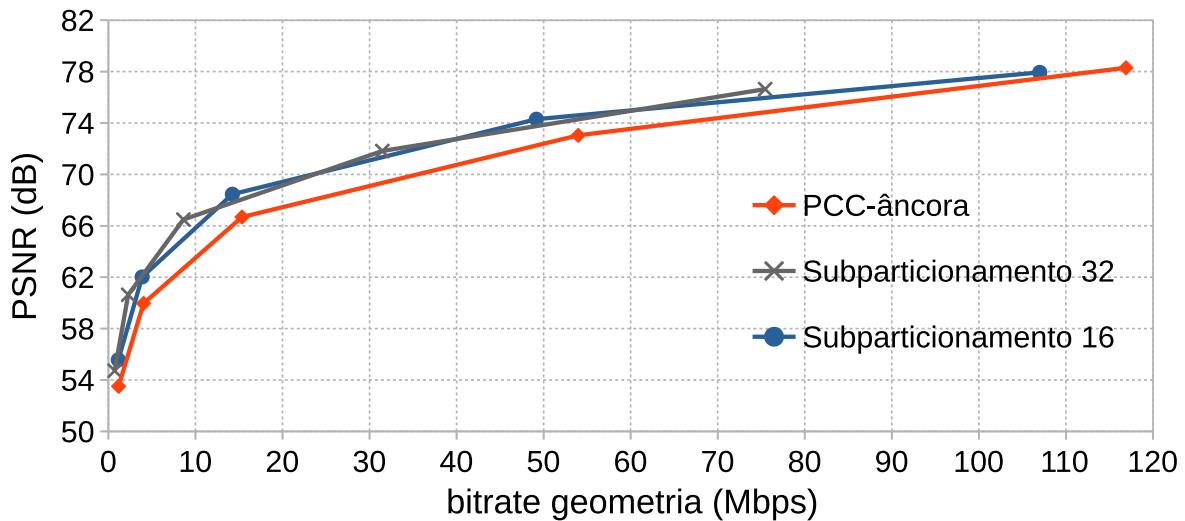


Figura 40: Gráfico RD de geometria para a proposta de subparticionamento.

Na Figura 41 são mostrados os resultados médios de qualidade e taxa para as nuvens testadas considerando a informação de textura. Os resultados médios mostram o Subparticionamento 16 tem uma curva RD superior as demais abordagens. O Subparticionamento 32 apresentou a maior redução de taxa, porém para os últimos *LoD* apresenta perdas significativas de qualidade em relação ao PCC-âncora. É importante destacar que essa perda de qualidade pode estar relacionada a falta de uma decisão de modo mais eficiente. Por outro lado, os resultados, em especial para o Subparticionamento 32, mostram que o esquema de subparticionamento é promissor para a obtenção de ganhos de redução de taxa.

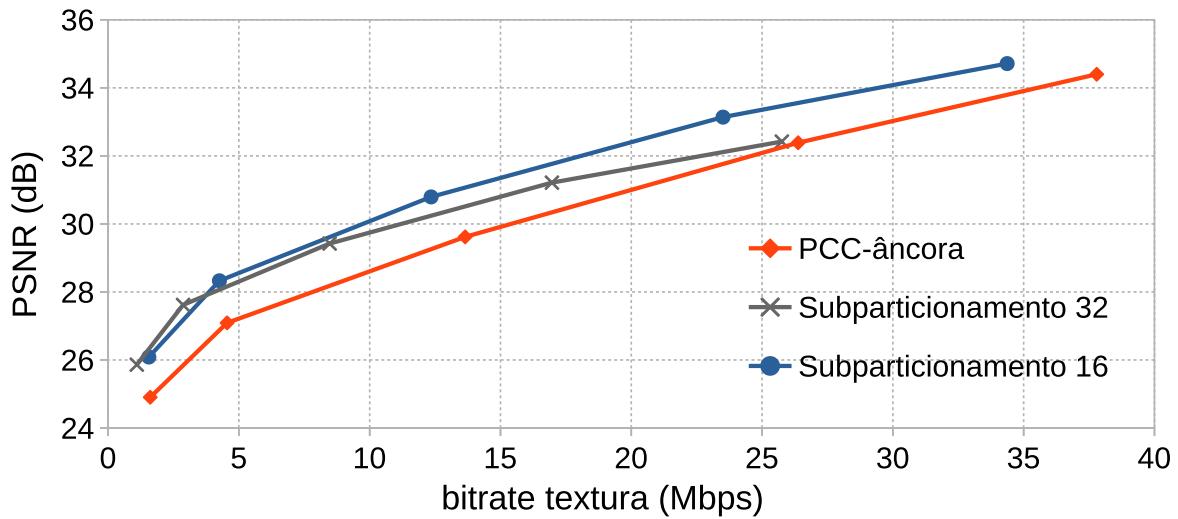


Figura 41: Gráfico RD de textura para a proposta de subparticionamento.

Na Tabela IX são mostrados os resultados percentuais médio de pontos codificados por meio da predição inter-nuvens. Os resultados apresentados mostram que a proposta de subparticionamento alcança o maior percentual de pontos inter para ambas as configurações, Subparticionamento 32 (Sub 32) e Subparticionamento 16 (Sub 16). Esses resultados são mais expressivos para o SUB 32, mas é importante salientar que a maior quantidade de blocos/pontos preditos como inter não significa maior eficiência de compressão, pois é necessária uma decisão de modo que garanta a melhor escolha tendo em vista qualidade e taxa.

Os resultados da métrica *Bjontegaard-Delta* (BD) para taxa de bits (BD-Rate) e PSNR (BD-PSNR) de geometria são apresentados na Tabela X. A métrica BD foi calculada para a solução proposta nas configurações subparticionamento 32 (Sub 32) e subparticionamento 16 (Sub 16), em relação ao PCC-âncora para cada uma das nuvens de pontos em g7, g8, g9 e g10. Na Tabela X o "geo.BD-PSNR" e o "geo.BD-RATE" referem-se ao BD-PSNR e ao BD-Rate das informações da geometria, considerando as abordagens P2Point (D1) e P2Plane (D2).

A partir dos resultados da Tabela X, é possível perceber que o esquema proposto obteve ganhos consideráveis em termos de eficiência de compressão para ambas as configurações de subparticionamento. Os resultados médios de BD-Rate de geometria mostram uma redução de 32,14% e 33,05% para D1 e D2 para a configuração Sub 32, em relação ao PCC-âncora. Os resultados da geometria BD-PSNR mostram ainda um aumento de 1,80 dB para D1 e 2,11 dB para D2.

Tabela IX: Percentual médio de pontos codificados por meio da predição inter-nuvens para os esquemas de subparticionamento de macroblocos em relação ao PCC-âncora.

Sequências	Abordagens	LoD				
		g6	g7	g8	g9	g10
<i>Long Dress</i>	<b>PCC</b>	0,00	0,10	1,51	4,21	5,35
	<b>Sub 16</b>	0,36	4,15	11,22	16,54	14,10
	<b>SUB 32</b>	94,84	96,04	93,67	86,00	69,70
<i>Loot</i>	<b>PCC</b>	0,98	9,60	13,98	11,75	23,38
	<b>Sub 16</b>	35,18	52,77	63,39	63,94	51,34
	<b>SUB 32</b>	97,67	98,04	97,01	93,25	84,14
<i>Red and Black</i>	<b>PCC</b>	1,07	5,34	7,66	13,81	15,79
	<b>Sub 16</b>	15,00	29,03	40,36	42,50	33,76
	<b>SUB 32</b>	96,38	96,03	93,96	88,34	75,49
<i>Soldier</i>	<b>PCC</b>	0,00	0,56	2,68	7,15	4,97
	<b>Sub 16</b>	4,00	12,96	29,07	48,92	51,31
	<b>SUB 32</b>	99,75	99,89	99,62	98,84	96,49
<i>Queen</i>	<b>PCC</b>	1,82	7,87	15,93	22,83	36,71
	<b>Sub 16</b>	24,48	51,90	68,87	77,16	76,66
	<b>SUB 32</b>	99,76	99,93	99,89	99,65	98,81
<b>Média</b>	<b>PCC</b>	<b>0,77</b>	<b>4,70</b>	<b>8,35</b>	<b>11,95</b>	<b>17,24</b>
	<b>Sub 16</b>	<b>15,80</b>	<b>30,16</b>	<b>42,58</b>	<b>49,81</b>	<b>45,43</b>
	<b>SUB 32</b>	<b>97,68</b>	<b>97,99</b>	<b>96,83</b>	<b>93,22</b>	<b>84,93</b>

Tabela X: Resultados de *Bjontegaard-Delta* de geometria para esquema de subparticionamento desenvolvido em relação ao PCC-âncora.

Sequência	geo.BD-RATE (%)				geo.BD-PSNR (dB)			
	Sub 16		Sub 32		Sub 16		Sub 32	
	D1	D2	D1	D2	D1	D2	D1	D2
<i>Long Dress</i>	-13,29	-31,47	-17,69	-12,55	0,73	2,14	0,87	0,62
<i>Loot</i>	-18,55	-29,92	-34,93	-37,42	0,98	1,89	1,89	2,29
<i>Red and Black</i>	-13,49	-21,81	-16,95	-12,24	0,71	1,35	0,79	0,56
<i>Soldier</i>	-17,64	-34,72	-49,66	-54,55	0,91	2,25	3,02	3,88
<i>Queen</i>	-21,89	-36,50	-41,49	-48,52	1,16	2,27	2,40	3,20
<b>Média</b>	<b>-16,97</b>	<b>-30,88</b>	<b>-32,14</b>	<b>-33,05</b>	<b>0,90</b>	<b>1,98</b>	<b>1,80</b>	<b>2,11</b>

Para a configuração de subparticionamento 16 (Sub 16), o BD-Rate de geometria obteve uma redução de 16,97% e 30,88% para D1 e D2, em relação ao PCC-âncora. Os resultados de BD-PSNR de geometria mostram aumento de 0,90 dB para D1 e 1,98 dB para D2. Neste sentido, os ganhos em BD-PSNR de Sub 16 foram um pouco inferiores em relação aos resultados obtidos pela configuração Sub 32.

Os resultados para textura são apresentados na Tabela XI. O "attr.BD-PSNR" e o "attr.BD-RATE" apresentam os valores de BD-PSNR e BD-Rate das informações de textura. Na Tabela XI, considerando a configuração Sub 32 a redução da taxa BD foi de 23,32%, 32,89% e 28,06% para os componentes Luminância (Luma), Crominância Azul (Cb) e Crominância Vermelha (Cr), respectivamente. Além disso, o BD-PSNR para textura apresentou ganho de 0,93 dB, 0,66 dB e 0,60 dB para os canais Luma, Cb e Cr, respectivamente.

Considerando a configuração Sub 16 também foram alcançados excelentes resultados, com uma redução da taxa de BD-Rate 31,77%, 11,30% e 13,92% para os canais Luma, Cb e Cr, respectivamente. Além disso, o BD-PSNR de textura apresentou um aumento de 1,40 dB, 0,21 dB e 0,29 dB para os mesmos canais.

Tabela XI: Resultados de *Bjontegaard-Delta* de textura para esquema de subparticionamento desenvolvido em relação ao PCC-âncora.

Sequências	Attr.BD-RATE (%)						Attr.BD-PSNR (dB)					
	Sub 16			Sub 32			Sub 16			Sub 32		
	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr
Long Dress	-30,04	-5,16	-7,28	7,06	-9,78	-0,99	1,28	0,08	0,12	-0,17	0,08	-0,04
Loot	-25,92	-11,00	-12,62	-22,54	-36,35	-38,48	0,91	0,24	0,29	0,54	0,88	0,98
Red and Black	-23,21	-8,99	-14,84	-6,83	-32,25	-17,08	0,88	0,15	0,29	0,05	0,54	0,22
Soldier	-37,70	-14,70	-13,68	-52,30	-50,90	-52,45	1,58	0,21	0,21	2,11	0,91	1,03
Queen	-41,97	-16,66	-21,16	-41,99	-35,15	-31,31	2,34	0,38	0,55	2,11	0,88	0,83
<b>Média</b>	<b>-31,77</b>	<b>-11,30</b>	<b>-13,92</b>	<b>-23,32</b>	<b>-32,89</b>	<b>-28,06</b>	<b>1,40</b>	<b>0,21</b>	<b>0,29</b>	<b>0,93</b>	<b>0,66</b>	<b>0,60</b>

Através desse experimento pode-se concluir que, o esquema de subparticionamento é promissor e pode trazer ganhos expressivos para eficiência de compressão de nuvens de pontos dinâmicas. Todavia, se faz necessário uma decisão de modo para realizar a escolha do melhor modo de predição e opção de tamanho de bloco, dessa forma viabilizando a obtenção uma melhor eficiência de codificação.

## 7.4 Resultados da Solução Completa

A solução completa proposta neste trabalho inclui o alinhamento das caixas delimitadoras (apresentado na Seção 6.1), a etapa de estimativa de movimento 3D, por meio do algoritmo DS (descrita na Seção 6.2), a decisão de modo (apresentada na Seção 6.5) e o subparticionamento que permite iniciar por macroblocos de tamanho  $32 \times 32 \times 32$  voxels ou  $16 \times 16 \times 16$  (conforme a Seção 6.4). Neste sentido, os resultados finais serão apresentados para estas duas configurações de tamanho máximo de macrobloco. Além disso, a discretização das coordenadas dos macroblocos, apresentada na Seção 6.3, também compõem esta solução.

Nas Figura 42 e Figura 43 são apresentadas as curvas RD, para informação de geometria, resultantes das simulações da solução completa com subparticionamento iniciando com macroblocos  $32 \times 32 \times 32$ , assim como subparticionamento iniciando com macroblocos de tamanho  $16 \times 16 \times 16$ , referenciados nos gráficos como Solução 32-16-8 e Solução 16-8, respectivamente. Além disso, são apresentados os resultados do PCC-âncora (MEKURIA; BLOM; CESAR, 2017), para serem confrontados com o esquema proposto. Nos gráficos RD os resultados de qualidade de geometria são apresentados por meio do PNSR P2Plane. Cada ponto das curvas representa um *LoD* de octree, onde o maior (g10), apresenta maior qualidade e consequentemente demanda maior taxa.

Os gráficos (a),(b) e (c) da Figura 42, mostram resultados individuais para três das cinco sequência de nuvens de pontos utilizadas. Por meio dos resultados dos gráficos de cada sequência, é possível concluir que os esquemas de predição proposto apresentam resultados superiores em relação ao PCC-âncora em ambas as configurações de subparticionamento, tanto em qualidade como em taxa de bits. Isso também pode ser visto nos gráficos (a) e (b) da Figura 43. Os resultados obtidos por ambas as configurações da solução proposta neste trabalho apresentam resultados muito similares, como pode ser visto pela sobreposição das curvas. No entanto, cabe destacar que nos gráficos (b) e (c) da Figura 42 e o gráfico (b) da Figura 43, a solução proposta de configuração 32-16-8 atinge uma menor taxa de bits e resultados de qualidade bastante semelhantes à solução 16-8. A solução 32-16-8 possui de modo geral ganhos em taxa. Porém, o preço pago por essa redução de taxa são perdas em qualidade mais notoriamente em g10 como é possível ver por meio do gráfico (b) da Figura

42 e o gráfico (a) da Figura 43. As diferenças entre as curvas RD tendem a ser menos perceptíveis em direção aos menores *LoD*.

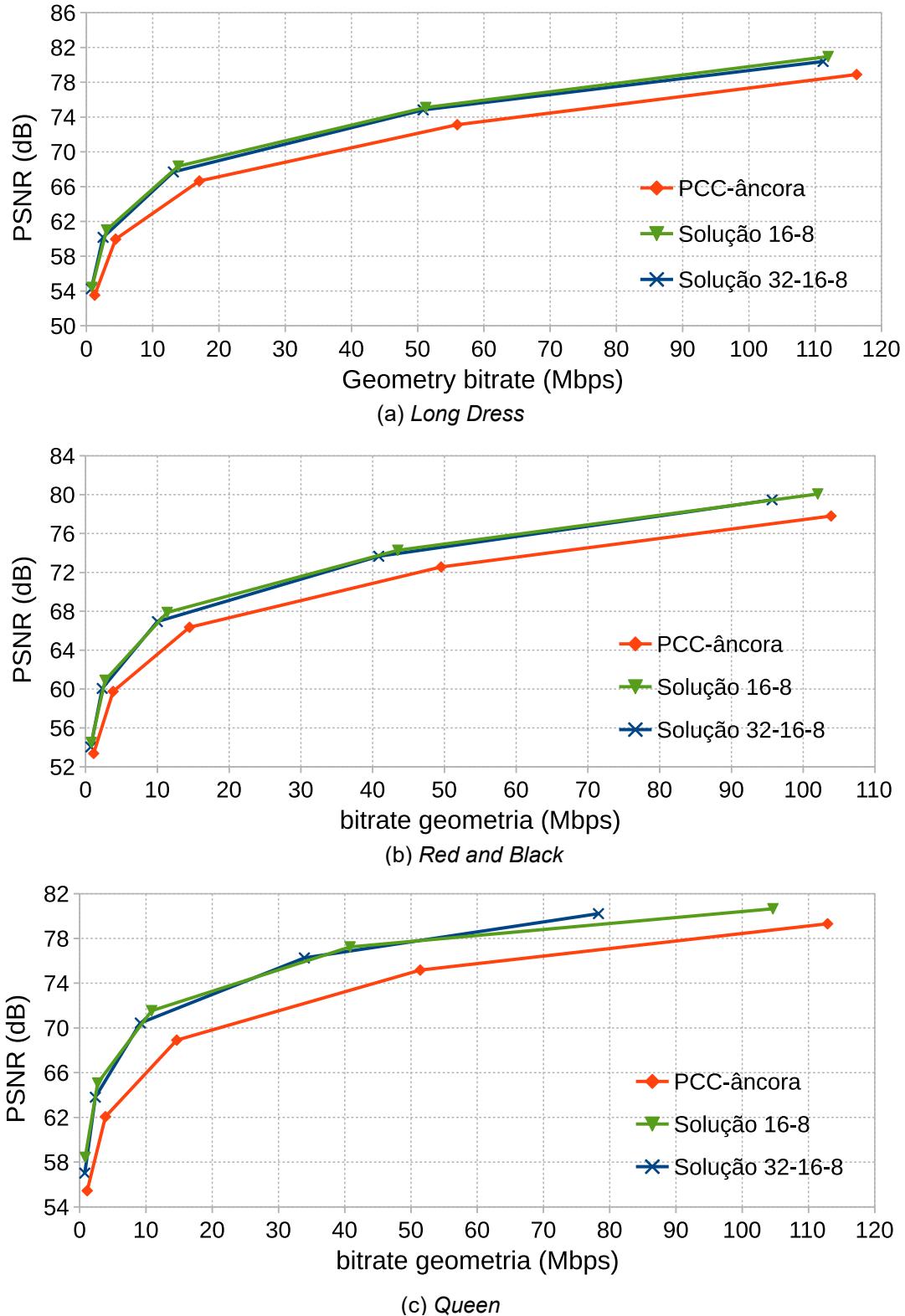


Figura 42: Gráficos RD para geometria.

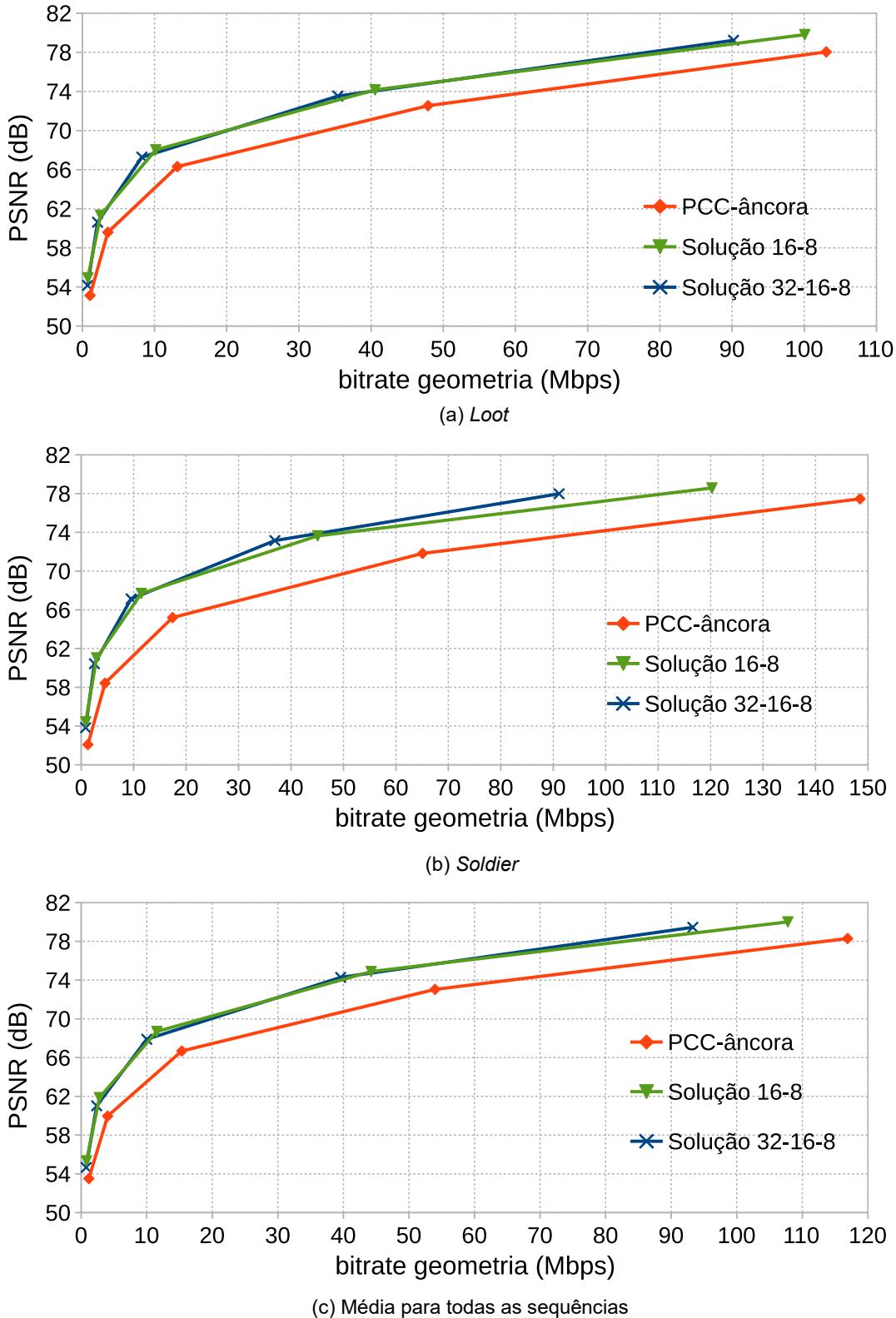


Figura 43: Gráficos RD para geometria.

Na Figura 43 (c) são apresentadas as curvas para os resultados médios de todas as sequências. É possível concluir que, a solução 16-8 consegue obter maior qualidade para os últimos *LoD*, mas também apresentam uma maior demanda de taxa de bits. Os resultados médios também comprovam a eficiência de codificação da solução proposta em relação ao PCC-âncora. Em especial a solução 32-16-8 atingem uma redução bastante significativa em taxa sem perdas em qualidade.

Na Figura 44 e 45 são apresentados gráficos RD para qualidade de textura e taxa de bits, onde os gráficos (a), (b) e (c) Figura 44 e os gráficos (a) e (b) da Figura 45 mostram resultados individuais para cada sequência de nuvens de pontos utilizadas. Esses gráficos comprovam que o esquema proposto também é muito superior ao PCC-âncora considerando a codificação de textura. Além disso, no gráfico (c) da Figura 45 são apresentados os resultados da média de todas as sequências para ambas as configurações, 32-16-8 e 16-8, em relação ao PCC-âncora. As configurações do esquema proposto apresentam resultados muito semelhantes, como pode ser visto pelas respectivas curvas RD. Por meio dos gráficos RD de textura fica ainda menos notória diferença entre a solução 32-16-8 e a solução 16-8. No entanto, no gráfico (c) da Figura 44 e o gráfico (b) da Figura 45 são mostrados melhores resultados obtidos pela solução proposta, onde é possível observar um aumento considerável de qualidade de textura com taxas muito inferiores ao PCC-âncora.

Resultados quantitativos mais detalhados são apresentados no apêndice A, Tabela XXIX (dados de geometria) e Tabela XXX (dados de textura). Além disso, no Apêndice B, da Figura 59 até à Figura 63 são apresentadas imagens de nuvens de pontos reconstruídas a partir da solução 32-16-8 em relação ao PCC-âncora.

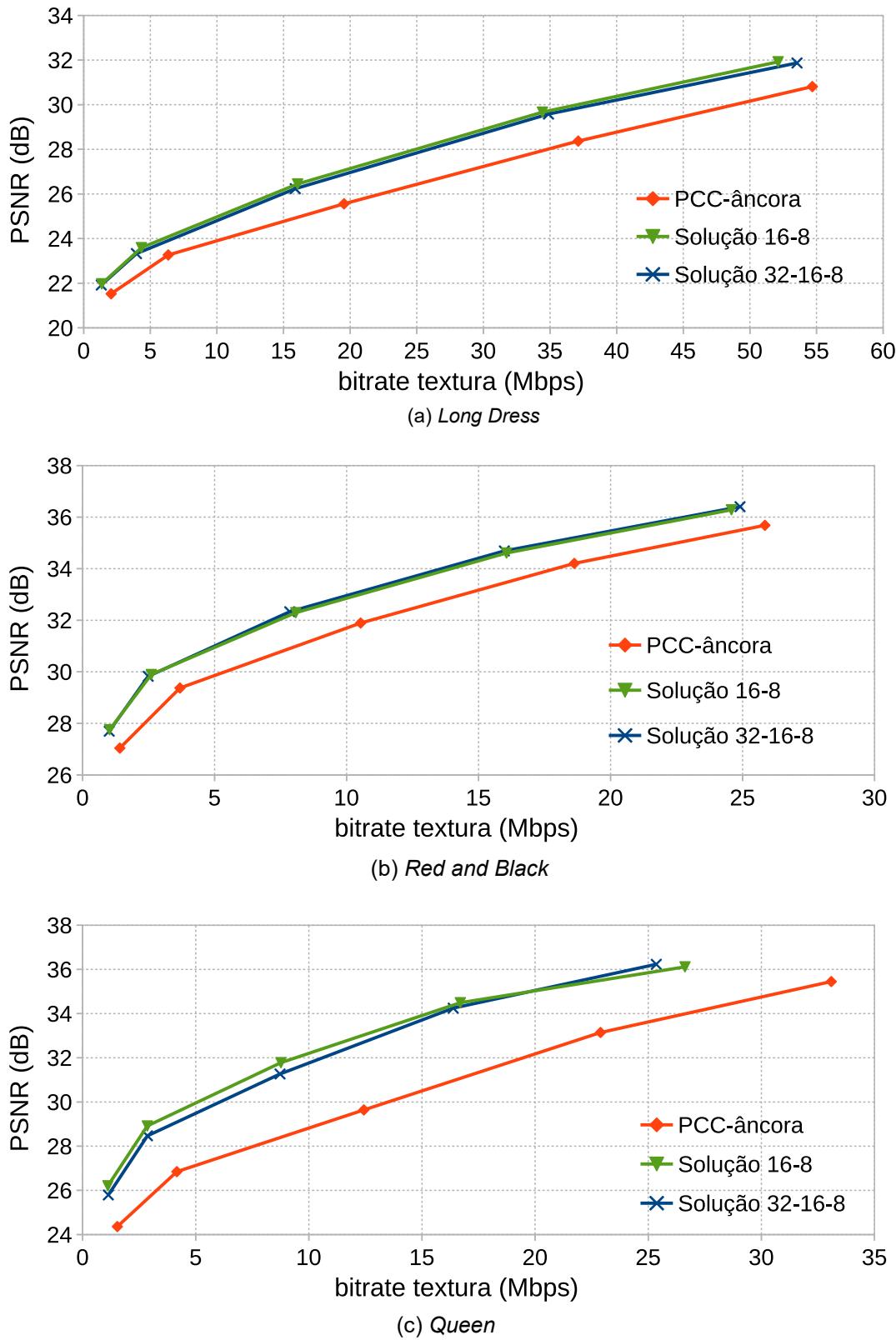


Figura 44: Gráficos RD para textura.

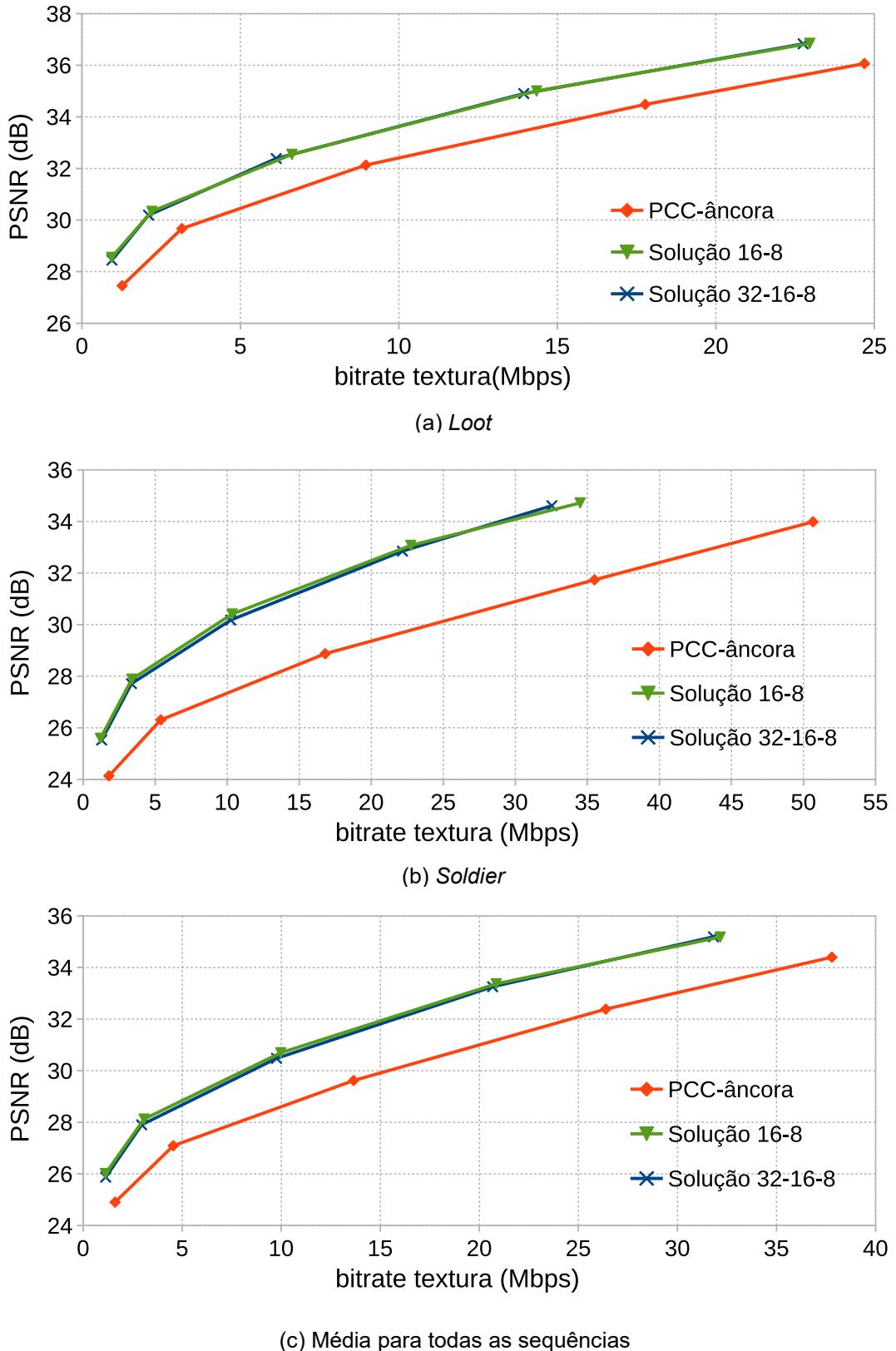


Figura 45: Gráficos RD para textura.

Na Tabela XII são mostrados os tempos de codificação em minutos para ambas as configurações da solução proposta Solução 32-16-8 e Solução 16-8. Os resultados de tempo são valores médios para codificar uma nuvem de pontos da sequência de nuvens de pontos. Na tabela são apresentados também os tempos médios de cada sequência avaliada em cada *LoD*. Os resultados de tempo de codificação mostram que a configuração de subparticionamento 16-8 (Sol. 16) consome em média um tempo 340% maior que o tempo da configuração 32-16-8 (Sol. 32) para g10, o que representa um custo 4 vezes maior. Para o menor nível de detalhes (g6), essa diferença é menor, mas ainda se mantém elevada, com um tempo 202% maior que a “Sol. 32”.

O alto custo computacional da solução de subparticionamento 16-8, ocorre devido a utilização de macroblocos de menor tamanho em relação a configuração 32-16-8. Dessa forma, há um número maior de comparações na ME para a solução 16-8. Isso ocorre porquê a ME é realizada somente sobre o maior macrobloco, ou seja, na solução 32-16-8 a ME é realizada somente sobre macroblocos de tamanho  $32 \times 32 \times 32$  voxels. Se for considerada a configuração g10, para as nuvens de pontos constituem as sequências utilizadas neste trabalho, em média existem 3800 blocos de tamanho  $32 \times 32 \times 32$  voxels. Se for considerado blocos de tamanho  $16 \times 16 \times 16$  voxels, o número médio de blocos passa para 16000, ou seja, são 4 vezes mais blocos a serem considerados pela ME. A relação de grandeza se assemelha ao resultado no aumento do tempo médio em g10 para a solução 16-8 em relação à solução 32-16-8, onde a solução 32-16-8 possui um tempo médio de 66 minutos para codificar uma nuvem de pontos enquanto a solução 16-8 demanda em torno de 291 minutos.

Na Tabela XII são ainda mostrados os tempos de codificação em minutos para o PCC. Como pode ser visto na tabela, a medida em que a qualidade aumenta (maior *LoD*) maior é a diferença de tempo de codificação da “Sol. 32” e a “Sol. 16” em relação ao PCC. Esses resultados eram esperados devido à complexidade das ferramentas adicionadas ao codificador, como ME em 3D e subparticionamento de macroblocos. Para o resultado de tempo médio das sequências avaliadas para g10, considerando a “Sol. 32”, a diferença atinge um aumento de aproximadamente 700 vezes o tempo consumido pelo PCC. Para o g6 a diferença é em torno de 130 vezes.

Tabela XII: Resultados de tempo (em minutos) para diferentes *LoD* das diferentes configurações do esquema proposto em relação ao PCC-âncora.

Sequências	<i>LoD</i>											
	g6		g7		g8		g9		g10			
	PCC	Sol.16	Sol.32	PCC	Sol.16	Sol.32	PCC	Sol.16	Sol.32	PCC	Sol.16	Sol.32
<i>Long</i>	0,004	1,30	0,43	0,007	5,26	1,72	0,02	22,39	10,59	0,04	64,38	29,53
<i>Loot</i>	0,004	0,86	0,35	0,009	3,53	0,99	0,03	15,35	4,64	0,07	53,97	16,48
<i>Red</i>	0,003	1,23	0,35	0,007	4,63	1,25	0,03	17,81	7,28	0,06	48,50	20,80
<i>Soldier</i>	0,005	1,57	0,42	0,008	4,26	0,78	0,02	14,79	2,94	0,06	68,22	13,76
<i>Queen</i>	0,003	1,21	0,40	0,007	3,75	0,71	0,02	15,30	2,63	0,06	53,70	10,71
<b>Média</b>	<b>0,003</b>	<b>1,23</b>	<b>0,39</b>	<b>0,007</b>	<b>4,28</b>	<b>1,09</b>	<b>0,024</b>	<b>17,12</b>	<b>5,61</b>	<b>0,05</b>	<b>57,75</b>	<b>18,25</b>
											<b>0,09 291,42 66,56</b>	

Devido à similaridade dos resultados apresentados pelos gráficos RD da Figura 42 à Figura 45 e com o custo computacional avaliado por meio do tempo consumido conforme apresentado na Tabela XII, é possível concluir que a configuração de sub-particionamento 32-16-8 apresenta a melhor relação entre custo computacional e eficiência de codificação, frente a configuração 16-8.

Os resultados da métrica *Bjontegaard-Delta* (BD) para taxa de bits (BD-Rate) e PSNR (BD-PSNR) de geometria são apresentados na Tabela XIII. A métrica BD foi calculada para a solução proposta nas configurações 32-16-8 e 16-8, em relação ao PCC-âncora para cada uma das nuvens de pontos em g7, g8, g9 e g10. Na Tabela XIII o "geo.BD-PSNR" e o "geo.BD-RATE" referem-se ao BD-PSNR e ao BD-Rate das informações da geometria, considerando as abordagens P2Point (D1) e P2Plane (D2).

Tabela XIII: Resultados de *Bjontegaard-Delta* de geometria da solução completa desenvolvida em relação ao PCC-âncora.

Sequência	geo.BD-RATE (%)				geo.BD-PSNR (dB)			
	Sub 16-8		Sub 32-16-8		Sub 16-8		Sub 32-16-8	
	D1	D2	D1	D2	D1	D2	D1	D2
<i>Long Dress</i>	-25,03	-38,11	-25,47	-33,89	1,41	2,66	1,4	2,26
<i>Loot</i>	-30,34	-41,75	-38,35	-43,74	1,68	2,77	2,21	2,92
<i>Red and Black</i>	-28,25	-39,18	-31,5	-36,38	1,57	2,62	1,77	2,36
<i>Soldier</i>	-39,70	-53,13	-49,31	-57,7	2,30	3,81	3,06	4,35
<i>Queen</i>	-43,92	-54,30	-45,07	-53,94	2,52	3,63	2,71	3,75
<b>Média</b>	<b>-33,45</b>	<b>-45,30</b>	<b>-37,94</b>	<b>-45,13</b>	<b>1,89</b>	<b>3,10</b>	<b>2,23</b>	<b>3,13</b>

A partir dos resultados da Tabela XIII, é possível perceber que o esquema proposto pode obter ganhos consideráveis em termos de eficiência de compressão para ambas as configurações de subparticionamento. Os resultados médios de BD-Rate de geometria mostram uma redução de 37,94% e 45,13% para D1 e D2 para a configuração 32-16-8, em relação ao PCC-âncora. Os resultados da geometria BD-PSNR mostram ainda um aumento de 2,23dB para D1 e 3,13 dB para D2.

Para a configuração de subparticionamento de 16-8, o BD-Rate de geometria obteve uma redução de 33,45% e 45,30% para D1 e D2, quando comparado os resultados do PCC-âncora. Os resultados de BD-PSNR de geometria mostram aumento de 1,89 dB para D1 e 3,10 dB para D2. Neste sentido, os ganhos em BD-PSNR foram um pouco inferiores em relação aos resultados obtidos pela configuração 32-16-8.

Os resultados para textura são apresentados na Tabela XIV. O "attr.BD-PSNR" e o "attr.BD-RATE" apresentam os valores de BD-PSNR e BD-Rate das informações de textura. Na Tabela XIV, considerando a configuração 32-16-8, para os atributos de textura, a redução da taxa BD foi de 41,55%, 34,85% e 33,08% para os componentes Luminância (Luma), Crominância Azul (Cb) e Crominância Vermelha (Cr), respectivamente. Além disso, o BD-PSNR para textura apresentou ganho de 1,90 dB, 0,69 dB e 0,72 dB para os canais Luma, Cb e Cr, respectivamente.

Considerando a configuração de subparticionamento 16-8, também foram alcançados excelentes resultados, com uma redução da taxa de BD-Rate 42,44%, 30,17% e 26,78% para os canais Luma, Cb e Cr, respectivamente. Além disso, o BD-PSNR de textura apresentou um aumento de 1,97 dB, 0,59 dB e 0,59 dB para os mesmos canais.

Tabela XIV: Resultados de *Bjontegaard-Delta* em textura para a solução completa desenvolvida em relação ao PCC-âncora.

Sequências	attr.BD-RATE (%)						attr.BD-PSNR (dB)					
	Sub 16-8			Sub 32-16-8			Sub 16-8			Sub 32-16-8		
	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr
Long	-34,29	-21,16	-24,28	-30,53	-20,27	-20,74	1,45	0,32	0,41	1,26	0,3	0,33
Loot	-35,22	-25,61	-27,99	-35,64	-33,64	-35,95	1,28	0,54	0,62	1,3	0,75	0,84
Red	-30,69	-16,53	4,62	-33,33	-30,7	-17,99	1,13	0,23	-0,09	1,25	0,5	0,32
Soldier	-58,15	-47,62	-45,84	-56,43	-49,43	-49,47	2,80	0,84	0,86	2,65	0,87	0,94
Queen	-53,83	-39,94	-40,40	-51,83	-40,21	-41,26	3,18	1,02	1,15	3,02	1,05	1,2
<b>Média</b>	<b>-42,44</b>	<b>-30,17</b>	<b>-26,78</b>	<b>-41,55</b>	<b>-34,85</b>	<b>-33,08</b>	<b>1,97</b>	<b>0,59</b>	<b>0,59</b>	<b>1,90</b>	<b>0,69</b>	<b>0,72</b>

Esta melhora na eficiência de codificação da solução proposta era esperada, uma vez que o PCC-âncora não explora de maneira eficiente ferramentas como a ME, nem mesmo apresenta blocos de tamanho variáveis.

A eficiência da solução de predição também pode ser vista pelo número de pontos preditos como inter. A Tabela XV mostra a porcentagem média de pontos preditos como inter para todas as cinco sequências de nuvens de pontos avaliadas, considerando o esquema proposto e o PCC-âncora para cada *LoD*.

Tabela XV: Número médio de pontos preditos como inter, separado por *LoD*.

<b><i>LoD</i></b>	<b>PCC- âncora (%)</b>	<b>Solução completa (%)</b>
<b>g6</b>	0,6	94,6
<b>g7</b>	4,7	87,5
<b>g8</b>	8,7	68,3
<b>g9</b>	11,6	51,9
<b>g10</b>	17,7	36,1

A solução proposta aumenta o número médio de pontos codificados como inter para todos os *LoD*, com uma variação de 94,6% a 36.1%, demonstrando ter uma predição inter-nuvens mais eficiente, suportada pela ME 3D e pelo subparticionamento de macroblocos.

A Figura 46 apresenta duas das cinco nuvens de pontos avaliadas, onde são destacados, em verde, os pontos preditos com a predição inter. Na Figura 46 (a) e (b) é apresentada a nuvem de pontos de número 1052 da sequência *Long Dress*, codificada com o PCC-âncora (a) e com a solução completa proposta com a configuração 32-16-8 (b) e *LoD* g10. Na Figura 46 (c) e (d) é apresentada a nuvem de ponto de número 1477 da sequência *Red and Black*, codificada com o PCC-âncora (c) e com a solução completa proposta com a configuração 32-16-8 (d). Por meio dos pontos em destaque conclui-se que a solução de predição proposta atinge uma maior porção da nuvem como inter, o que favorece a redução da taxa de bits da nuvem codificada.

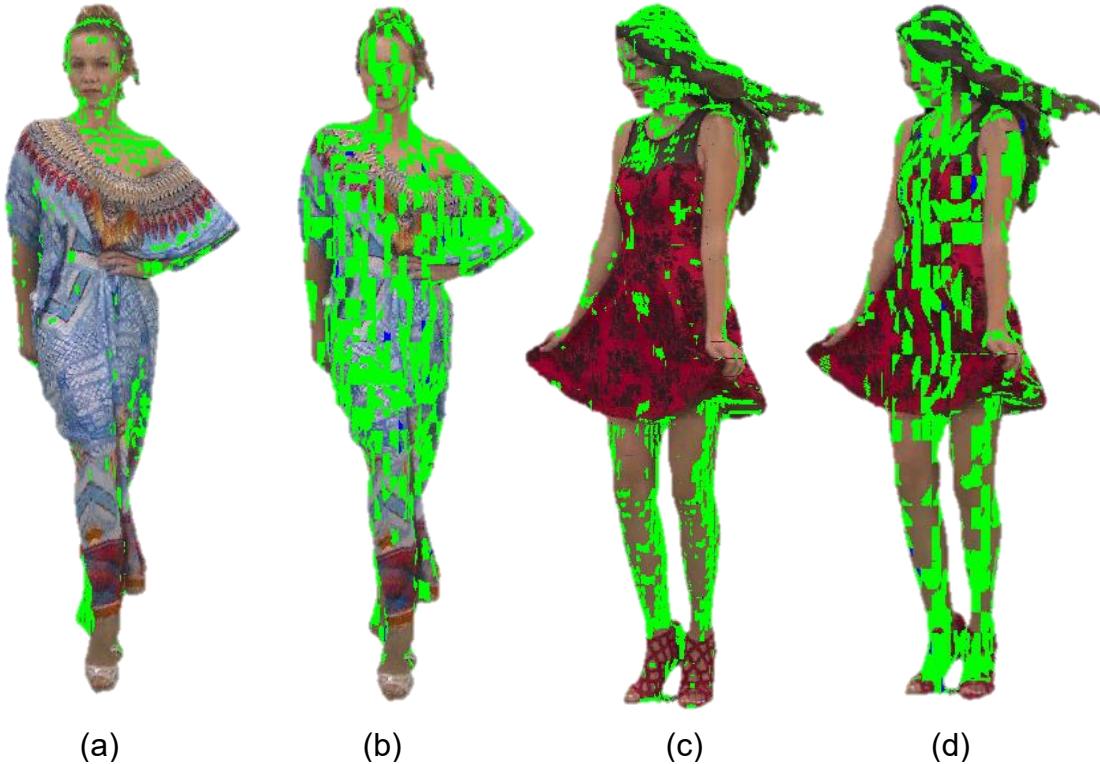


Figura 46: Avaliação visual da quantidade de blocos preditos com a predição inter (blocos em verde).

Conforme resultados apresentados para a solução completa do esquema de predição desenvolvido neste trabalho, é possível concluir que a solução proposta atingiu ganhos relevantes em eficiência de compressão em relação ao PCC-âncora. Além disso, os resultados individuais para o esquema de alinhamento das caixas delimitadoras, apresentados na Seção 7.2, assim como o subparticionamento de macroblocos discutidos na Seção 7.3, apresentaram ganhos relevantes em eficiência de codificação que contribuíram com a solução final.

## 8 CONCLUSÕES

Este trabalho apresentou uma exploração de novos esquemas para aumentar a eficiência da predição na compressão de nuvem de pontos dinâmicas. A solução completa utiliza um esquema de alinhamento das caixas delimitadoras e também emprega uma estimativa de movimento (ME) 3D baseada em macroblocos, bem como um esquema de subparticionamento de em blocos menores. A solução proposta permite o uso de macroblocos com tamanho máximo de 32x32x32 ou 16x16x16, e sub-particionamento de 32-16-8 ou 16-8, respectivamente. Neste trabalho também foi apresentado um algoritmo de decisão de modo, que define o melhor modo de codificação de cada bloco, levando em conta a eficiência de codificação tanto das informações de geometria quanto de textura.

Além disso, a revisão da literatura realizada demonstrou que, até o momento da elaboração deste texto, não há nenhum trabalho publicado com solução que utilize subparticionamento de macroblocos e tão pouco uma solução de ME em 3D para compressão de nuvens de pontos dinâmicas.

Neste sentido, o trabalho apresentou uma avaliação de correlação temporal de nuvens de pontos, como objetivo de verificar as regiões de maior ocorrência de blocos similares. Na sequência, foram implementados e adaptados dois algoritmos clássicos de ME utilizados em vídeos 2D, o *Diamond Search* (DS) e o *N-Step-Search* (NSS). Assim esses algoritmos foram avaliados sobre a perspectiva de informação de geometria 3D. A partir dessa avaliação o DS foi selecionado para constituir o esquema de predição proposto neste trabalho.

O esquema de predição proposto, empregou uma estimativa de movimento 3D por meio de uma versão adaptada do algoritmo DS, com a possibilidade de operação sobre macroblocos de tamanho 32x32x32 ou ainda sobre macroblocos de tamanho 16x16x16.

Além disso, uma abordagem de subparticionamento foi proposta, onde viabiliza o particionamento de um macrobloco  $32 \times 32 \times 32$  em blocos  $16 \times 16 \times 16$  e  $8 \times 8 \times 8$ , que podem ser preditos individualmente com os modos inter ou intra. Uma decisão de modo também foi desenvolvida para definir a melhor opção de codificação, levando em consideração a melhor relação entre a taxa de bits e a distorção.

As avaliações foram realizadas para as cinco sequências de nuvens de pontos que contemplam as condições comuns de testes (CTC) propostas pelo MPEG. Para cada sequência cinco níveis de detalhes da *octree* foram avaliados, onde esses variam de 6 (g6) a 10 bits (g10). Os resultados objetivos de qualidade foram gerados usando as abordagens ponto-a-ponto (P2P) e ponto-a-plano (P2Plane) por meio de PSNR e MSE, seguindo as CTC.

Os resultados mostram que o esquema proposto é capaz de obter redução no BD-Rate de aproximadamente 41% para textura e até 45% para a informação de geometria, quando comparado com o codificador PCC-Âncora. Além disso, o esquema proposto obteve resultados superiores de qualidade em comparação ao PCC-âncora, com aumento médio de BD-PSNR de 3,13 dB em geometria e 1,9 dB na textura (para o canal de luminância).

## 8.1 Trabalhos Futuros

Algumas linhas de pesquisas já estão sendo desenvolvidas como continuação deste trabalho. Entre as possibilidades a serem desenvolvidas cabe destacar a exploração de diferentes configurações de grupo de nuvens de pontos (*group of clouds* - GoC), já que os resultados desta proposta tratam somente GoC de tamanho 2. Assim, o uso de um GoC maior pode beneficiar a redução na taxa de bits já que, dessa forma, um maior número de nuvens de pontos pode ser codificado com predição inter-nuvens.

Ferramenta como filtro de deblocagem também pode ser uma abordagem explorada em trabalhos futuros. Como pode ser visto através de algumas imagens reconstruídas a partir da compressão de nuvem de pontos dinâmica com aplicação da ME 3D, são gerados artefatos principalmente nas regiões de bordas das nuvens de pontos, gerados possivelmente por pontos de blocos de diferentes números de *voxels*. Dessa forma, uma abordagem de filtragem desses pontos poderia contribuir para

geração de nuvens de pontos com melhores aspectos tanto em qualidade subjetiva como também de qualidade objetiva.

A compressão dos atributos de textura por meio do PCC-âncora, utiliza o algoritmo JPEG clássico. Uma contribuição interessante pode ser a utilização de algoritmos ou codificadores de textura mais eficientes, como até mesmo a aplicação do HEVC para codificar somente textura. Essa adequação para o codificador pode contribuir tanto para redução de taxa como para uma melhor qualidade de textura para as nuvens de pontos reconstruídas.

A adequação e aplicação da ME 3D sugerido neste trabalho, também pode ser aplicada no codificador V-PCC proposto atualmente pelo MPEG. O codificador V-PCC utiliza somente informações das nuvens planificadas para realizar a ME. Neste contexto, uma ME 3D pode melhorar a eficiência de compressão para essa abordagem, diante da exploração mais específica para esse tipo de conteúdo.

Estudos mais aprofundados em relação as regiões de maior ocorrência de blocos selecionados na ME, visto por meio dos mapas de calor, podem auxiliar no desenvolvimento de algoritmo de ME mais otimizado para os possíveis padrões de ocorrências. Além disso, técnicas de *machine learning* podem contribuir para a redução de complexidade da ME, principalmente a partir do treinamento de redes neurais em função das posições das maiores ocorrência de blocos correlacionados, e assim viabilizar melhores resultados de qualidade.

## REFERÊNCIAS

ADAMI, Z.; TREGLIA, D. Quaternion compression. Game Programming Gems, 2002, 187-191.

AFONSO, Vladimir, et al. Energy-Aware Motion and Disparity Estimation System for 3D-HEVC With Run-Time Adaptive Memory Hierarchy. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 29.6: 1878-1892.

AINALA, Khartik, et al. An improved enhancement layer for octree based point cloud compression with plane projection approximation. In: Applications of Digital Image Processing XXXIX. International Society for Optics and Photonics, 2016. p. 99710R.

ALEXIOU, E.; EBRAHIMI, T. On subjective and objective quality evaluation of point cloud geometry. In: INTERNATIONAL CONFERENCE ON QUALITY OF MULTIMEDIA EXPERIENCE (QOMEX 2017), Erfurt, 2017, pp. 1-3.

ARMENI, I. et al. 3D Semantic Parsing of Large-Scale Indoor Spaces. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016. Las Vegas, NV, 2016, pp. 1534-1543.

BRIDGET. BRIDging the Gap for Enhanced broadcasT (Bridget) European research project.

CAO, Chao; PREDA, Marius; ZAHARIA, Titus. 3D Point Cloud Compression: A Survey. In: The 24th International Conference on 3D Web Technology. 2019. p. 1-9.

C3DC. Culture 3D Clouds: French National research project.

CHEN, Yue; MUKHERJEE, Debangha. Variable block-size overlapped block motion compensation in the next generation open-source video codec. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 938-942.

CHEN, Jianle, et al. The Joint Exploration Model (JEM) for Video Compression with Capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology, 2019.

CHEN, Siyuan; F LAEFER, Debra; MANGINA, Eleni. State of technology review of civilian UAVs. Recent Patents on Engineering, v. 10, n. 3, p. 160-174, 2016.

CORREA, G.; ASSUNCAO, P.; AGOSTINI, L.; SILVA CRUZ, L. A. da. 2015. Complexity-Aware High Efficiency Video Coding (1st. ed.). Springer Publishing Company, Incorporated.

DOREA, C.; QUEIROZ, R. L. de. Block-Based Motion Estimation Speedup for Dynamic Voxelized Point Clouds. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 201825th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 2964-2968.

DOU, Mingsong, et al. 3D scanning deformable objects with a single RGBD sensor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. p. 493-501.

D'EON, E., et al. 8i voxelized full bodies, version 2—a voxelized point cloud dataset. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m40059 M, 2017, 74006.

GHANBARI, Mohammed. Standard codecs: Image compression to advanced video coding. Iet, 2003.

HE, Lanyi; ZHU, Wenjie; XU, Yiling. Best-effort projection based attribute compression for 3d point cloud. In: 2017 23rd Asia-Pacific Conference on Communications (APCC). IEEE, 2017. p. 1-6.

HOPPE, Hugues, et al. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on Computer graphics and interactive techniques. 1992. p. 71-78.

HUANG, Yan, et al. Octree-Based Progressive Geometry Coding of Point Clouds. SPBG, 2006, 6: 103-110.

HUANG, Yan, et al. A generic scheme for progressive point cloud coding. IEEE Transactions on Visualization and Computer Graphics, 2008, 14.2: 440-453.

JAVAHERI, Alireza, et al. Subjective and objective quality evaluation of 3D point cloud denoising algorithms. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2017. p. 1-6.

JING, Xuan; CHAU, Lap-Pui. An efficient three-step search algorithm for block motion estimation. IEEE transactions on multimedia, 2004, 6.3: 435-438.

- KAMMERL, Julius, et al. Real-time compression of point cloud streams. In: 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012. p. 778-785.
- KATHARIYA, Birendra, et al. Embedded binary tree for dynamic point cloud geometry compression with graph signal resampling and prediction. In: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2017. p. 1-4.
- KOGA, Toshio. Motion compensated interframe coding for video-conferencing. In: Proc. Nat. Telecommun. Conf. 1981. p. G5. 3.1-G5. 3.5.
- LI, Li, et al. Advanced 3D Motion Prediction for Video Based Point Cloud Attributes Compression. In: 2019 Data Compression Conference (DCC). IEEE, 2019. p. 498-507.
- LI, Xiang; LI, Eric Q.; CHEN, Yen-Kuang. Fast multi-frame motion estimation algorithm with adaptive search strategies in H. 264. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 2004. p. iii-369.
- LOOP, Charles, et al. Microsoft voxelized upper bodies-a voxelized point cloud dataset. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012, 2016.
- LOOP, Charles; ZHANG, Cha; ZHANG, Zhengyou. Real-time high-resolution sparse voxelization with application to image-based modeling. In: Proceedings of the 5th High-Performance Graphics Conference. 2013. p. 73-79.
- MAMMOU, K. PCC Test Model Category 2 V1. ISO/IEC JTC1/SC29/WG11 N17348, 2018.
- MARTIN, G. Range encoding: an algorithm for removing redundancy from a digitised message. In: Video and Data Recording Conference, Southampton, 1979. 1979. p. 24-27.
- MELO, Mateus, et al. Objective and Subjective Video Quality Assessment in Mobile Devices for Low-Complexity H. 264/AVC Codecs. In: Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web. 2017. p. 429-432.
- MEKURIA, Rufael; BLOM, Kees; CESAR, Pablo. Design, implementation, and evaluation of a point cloud codec for tele-immersive video. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 27.4: 828-842.
- MEKURIA, R.; LI, Z.; TULVAN, C. Evaluation criteria for PCC (Point Cloud Compression). MPEG2016 n16332, Geneva, February 2016.

MEKURIA, Rafael; CESAR, Pablo. MP3DG-PCC, open source software framework for implementation and evaluation of point cloud compression. In: Proceedings of the 24th ACM international conference on Multimedia. 2016. p. 1222-1226.

MORELL, Vicente, et al. Geometric 3D point cloud compression. Pattern Recognition Letters, 2014, 50: 55-62.

MORENO, Carlos; CHEN, Yilin; LI, Ming. A dynamic compression technique for streaming kinect-based Point Cloud data. In: 2017 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2017. p. 550-555.

GARCIA, Diogo C.; DE QUEIROZ, Ricardo L. Context-based octree coding for point-cloud video. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 1412-1416.

GARCIA, Diogo C.; DE QUEIROZ, Ricardo L. Intra-frame context-based octree coding for point-cloud geometry. In: 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018. p. 1807-1811.

GOLLA, Tim; KLEIN, Reinhard. Real-time point cloud compression. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015. p. 5087-5092.

GONÇALVES, Mateus, et al. Encoding Efficiency and Computational Cost Assessment of State-Of-The-Art Point Cloud Codecs. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019. p. 3726-3730.

GONCALVES, Paulo, et al. Complexity-Aware TZS Algorithm for Mobile Video Encoders. Journal of Integrated Circuits and Systems, 2019, 14.3: 1-9.

PANDEY, Gaurav; MCBRIDE, James R.; EUSTICE, Ryan M. Ford campus vision and lidar data set. The International Journal of Robotics Research, 2011, 30.13: 1543-1552.

PORTO, Marcelo, et al. A high throughput and low cost diamond search architecture for HDTV motion estimation. In: 2008 IEEE International Conference on Multimedia and Expo. IEEE, 2008. p. 1033-1036.

PORTO, M., et al. Investigation of motion estimation algorithms targeting high resolution digital video compression. In: ACM Brazilian Symposium on Multimedia and Web. 2007. p. 111-118.

DE QUEIROZ, Ricardo L.; CHOU, Philip A. Compression of 3d point clouds using a region-adaptive hierarchical transform. IEEE Transactions on Image Processing, 2016, 25.8: 3947-3956.

DE QUEIROZ, Ricardo L.; CHOU, Philip A. Motion-compensated compression of dynamic voxelized point clouds. *IEEE Transactions on Image Processing*, 2017, 26.8: 3886-3895.

QUEIROZ, Ricardo L.; CHOU, Philip A. Compression of 3d point clouds using a region-adaptive hierarchical transform. *IEEE Transactions on Image Processing*, 2016, 25.8: 3947-3956.

RAO, K. R.; HWANG, J. J.; KIM, D. N. *High Efficiency Video Coding and other emerging standards*. River publishers, 2017.

RUSU, Radu Bogdan; COUSINS, Steve. 3d is here: Point cloud library (pcl). In: 2011 IEEE international conference on robotics and automation. IEEE, 2011. p. 1-4.

SALOMON, David. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

SANTOS, Cristiano F., et al. A Sub-Partitioning Method for Point Cloud Inter-prediction Coding. In: 2018 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2018. p. 1-4.

SCHNABEL, Ruwen; KLEIN, Reinhard. Octree-based Point-Cloud Compression. Spbg, 2006, 6: 111-120.

SHAO, Yiting, et al. Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform. In: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2017. p. 1-4.

SCHWARZ, Sebastian, et al. Common test conditions for point cloud compression. Document ISO/IEC JTC1/SC29/WG11 w17766, Ljubljana, Slovenia, 2018.

SCHWARZ, Sebastian, et al. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2018, 9.1: 133-148.

SHI, Yun Q.; SUN, Huifang. *Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards*. CRC press, 1999.

SMITH, Jason; PETROVA, G.; SCHAEFER, Scott. Progressive encoding and compression of surfaces generated from point cloud data. *Computers & Graphics*, 2012, 36.5: 341-348.

SULLIVAN, Gary J., et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 2012, 22.12: 1649-1668.

SULLIVAN, Gary J.; WIEGAND, Thomas. Video compression-from concepts to the H. 264/AVC standard. *Proceedings of the IEEE*, 2005, 93.1: 18-31.

THANOU, Dorina; CHOU, Philip A.; FROSSARD, Pascal. Graph-based motion estimation and compensation for dynamic 3D point cloud compression. In: 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015. p. 3235-3239.

THANOU, Dorina; CHOU, Philip A.; FROSSARD, Pascal. Graph-based compression of dynamic 3D point cloud sequences. *IEEE Transactions on Image Processing*, 2016, 25.4: 1765-1778.

TIAN, D. et al. Evaluation Metrics for Point Cloud Compression. *MPEG2016 m39316*, China, October 2016.

TIAN, Dong, et al. Geometric distortion metrics for point cloud compression. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. p. 3460-3464.

TULVAN, R.; MEKURIA, R.; LI, Z. Use Cases for Point Cloud Compression (Point Cloud Compression). *MPEG2016 n16332*, Geneva, February 2016.

WALLACE, Gregory K. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 1992, 38.1: xviii-xxxiv.

ZAKHARCHENKO, V. Algorithm description of mpeg-pcc-tmc2. In: ISO/IEC JTC1/SC29/WG11/N17767. 2018.

ZAKHARCHENKO, V.; KIM, J. Point Cloud Compression Exploration Experiment 2.1 on 3D motion estimation. ISO/IEC JTC1/SC29/WG11 w18043. 2018.

ZHANG, Cha; FLORENCIO, Dinei; LOOP, Charles. Point cloud attribute compression with graph transform. In: 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014. p. 2066-2070.

ZHANG, Cha; FLORÊNCIO, Dinei. Analyzing the optimality of predictive transform coding using graph-based models. *IEEE Signal Processing Letters*, 2012, 20.1: 106-109.

ZHU, Shan; MA, Kai-Kuang. A new diamond search algorithm for fast block-matching motion estimation. *IEEE transactions on Image Processing*, 2000, 9.2: 287-290.



# APÊNDICE A – RESULTADOS DE EXPERIMENTOS NA AVALIAÇÃO DOS ALGORITMOS DE ME DESENVOLVIDOS

Este apêndice apresenta os resultados completos da avaliação dos algoritmos de ME apresentados, de modo resumido, na Seção 7.

A tabelas que segue apresenta resultados de experimentos do FS em relação ao 4SS e DS para macroblocos que tamanho 32.

Tabela XVI: Avaliação dos algoritmos de ME para macroblocos de tamanho 32.

Sequências	P2plane			P2p		
	FS15	4SS	DS	FS15	4SS	DS
<i>Long Dress</i>	68,47	67,31	66,55	59,85	56,87	58,24
<i>Loot</i>	74,10	71,96	71,13	69,49	63,26	65,18
<i>Red and Black</i>	69,18	66,04	65,73	62,48	56,93	58,39
<i>Soldier</i>	75,62	73,90	75,62	71,62	69,39	71,63
<b>Média</b>	<b>71,84</b>	<b>69,80</b>	<b>69,76</b>	<b>65,86</b>	<b>61,61</b>	<b>63,36</b>

A Tabela XVII apresenta resultados de experimentos do FS em relação ao 4SS e DS para macroblocos que tamanho 16.

Tabela XVII: Avaliação dos algoritmos de ME para macroblocos de tamanho 16.

Sequências	P2plane			P2p		
	FS15	TSS	DS	FS15	TSS	DS
<i>Long Dress</i>	71,64	66,94	64,85	61,44	54,84	53,04
<i>Loot</i>	76,03	72,29	71,33	71,67	62,55	61,66
<i>Red and Black</i>	71,61	65,88	64,40	69,18	56,18	55,11
<i>Soldier</i>	76,44	74,28	76,15	72,18	68,77	71,84
<b>Média</b>	<b>73,93</b>	<b>69,84</b>	<b>69,18</b>	<b>68,62</b>	<b>60,59</b>	<b>60,41</b>

A Tabela XVIII apresenta resultados de experimentos do FS em relação ao 4SS e DS sem a utilização do algoritmo ICP para macroblocos que tamanho 32.

Tabela XVIII: Avaliação dos algoritmos de ME SEM ICP para macroblocos de tamanho 32.

Sequências	P2plane			P2p		
	FS15	TSS	DS	FS15	TSS	DS
<i>Long Dress</i>	65,24	58,93	61,94	60,40	53,66	51,22
<i>Loot</i>	75,55	72,62	72,35	71,72	69,26	66,67
<i>Red and Black</i>	67,89	63,00	63,88	61,98	57,45	55,76
<i>Soldier</i>	76,01	75,87	75,36	72,07	71,96	71,57
<b>Média</b>	<b>71,17</b>	<b>67,60</b>	<b>68,38</b>	<b>66,54</b>	<b>63,08</b>	<b>61,30</b>

A Tabela XIX apresenta resultados de experimentos do FS em relação ao 4SS e DS sem a utilização do algoritmo ICP para macroblocos que tamanho 16.

Tabela XIX: Avaliação dos algoritmos de ME SEM ICP para macroblocos de tamanho 16.

Sequências	P2plane			P2p		
	FS15	TSS	DS	FS15	TSS	DS
<i>Long Dress</i>	65,24	58,93	61,94	60,40	53,66	51,22
<i>Loot</i>	75,55	72,62	72,35	71,72	69,26	66,67
<i>Red and Black</i>	67,89	63,00	63,88	61,98	57,45	55,76
<i>Soldier</i>	76,01	75,87	75,36	72,07	71,96	71,57
<b>Média</b>	<b>71,17</b>	<b>67,60</b>	<b>68,38</b>	<b>66,54</b>	<b>63,08</b>	<b>61,30</b>

A Tabela XX apresenta resultados comparativos de tempo do FS e do DS para macroblocos de tamanho 32x32x32 com e sem ICP.

Tabela XX: Avaliação comparativa de tempo (minutos) dos algoritmos entre FS e DS com ICP e tamanho de macroblocos 32x32x32 voxels em comparação aos respectivos algoritmos sem o ICP.

Sequências	Sem ICP		Com ICP	
	FS M32	DS M32	FS M32	DS M32
<i>Long Dress</i>	1247	44	6161	89
<i>Loot</i>	1387	47	7922	60
<i>Red and Black</i>	1124	53	7212	72
<i>Soldier</i>	1864	61	12386	71
<b>Média</b>	<b>1406</b>	<b>51</b>	<b>8420</b>	<b>73</b>

A Tabela XX apresenta resultados comparativos de tempo do FS e do DS para macroblocos de tamanho 16x16x16 com e sem ICP.

Tabela XXI: Avaliação comparativa de tempo (minutos) dos algoritmos entre FS e DS com ICP e tamanho de macroblocos 16x16x16 voxel/s em comparação aos respectivos algoritmos sem o ICP.

Sequências	Sem ICP		Com ICP	
	FS M32	DS M32	FS M32	DS M32
<i>Long Dress</i>	1328	161	3701	172
<i>Loot</i>	1471	145	4741	167
<i>Red and Black</i>	1382	153	3542	246
<i>Soldier</i>	2520	248	6941	325
<b>Média</b>	<b>1675</b>	<b>177</b>	<b>4731</b>	<b>228</b>

A Tabela XXII apresenta resultados comparativos de qualidade de geometria do FS e do DS para macroblocos de tamanho 16x16x16 com e sem ICP.

Tabela XXII: Avaliação comparativa de qualidade de geometria dos algoritmos entre FS e DS com ICP e também sem ICP para macroblocos 16x16x16.

Sequências	P2plane				P2p			
	FS15		DS		FS15		DS	
	Com ICP	Sem ICP						
<i>Long Dress</i>	68,47	62,78	65,34	62,33	59,85	60,43	56,11	56,94
<i>Loot</i>	74,10	73,66	72,85	71,54	69,49	70,57	68,75	68,42
<i>Red and Black</i>	69,18	65,21	66,58	63,23	62,48	62,14	59,02	59,35
<i>Soldier</i>	75,76	74,88	75,61	74,88	71,62	71,25	71,63	71,25
<b>Média</b>	<b>71,88</b>	<b>69,13</b>	<b>70,10</b>	<b>67,99</b>	<b>65,86</b>	<b>66,10</b>	<b>63,88</b>	<b>63,99</b>

A Figura 47 apresenta os mapas de calor gerados para o algoritmo FS com macroblocos de tamanho 16x16x16.

Na Figura 48 são apresentados os mapas de calor gerados para o algoritmo FS com macroblocos de tamanho 32x32x32 sem o uso do ICP.

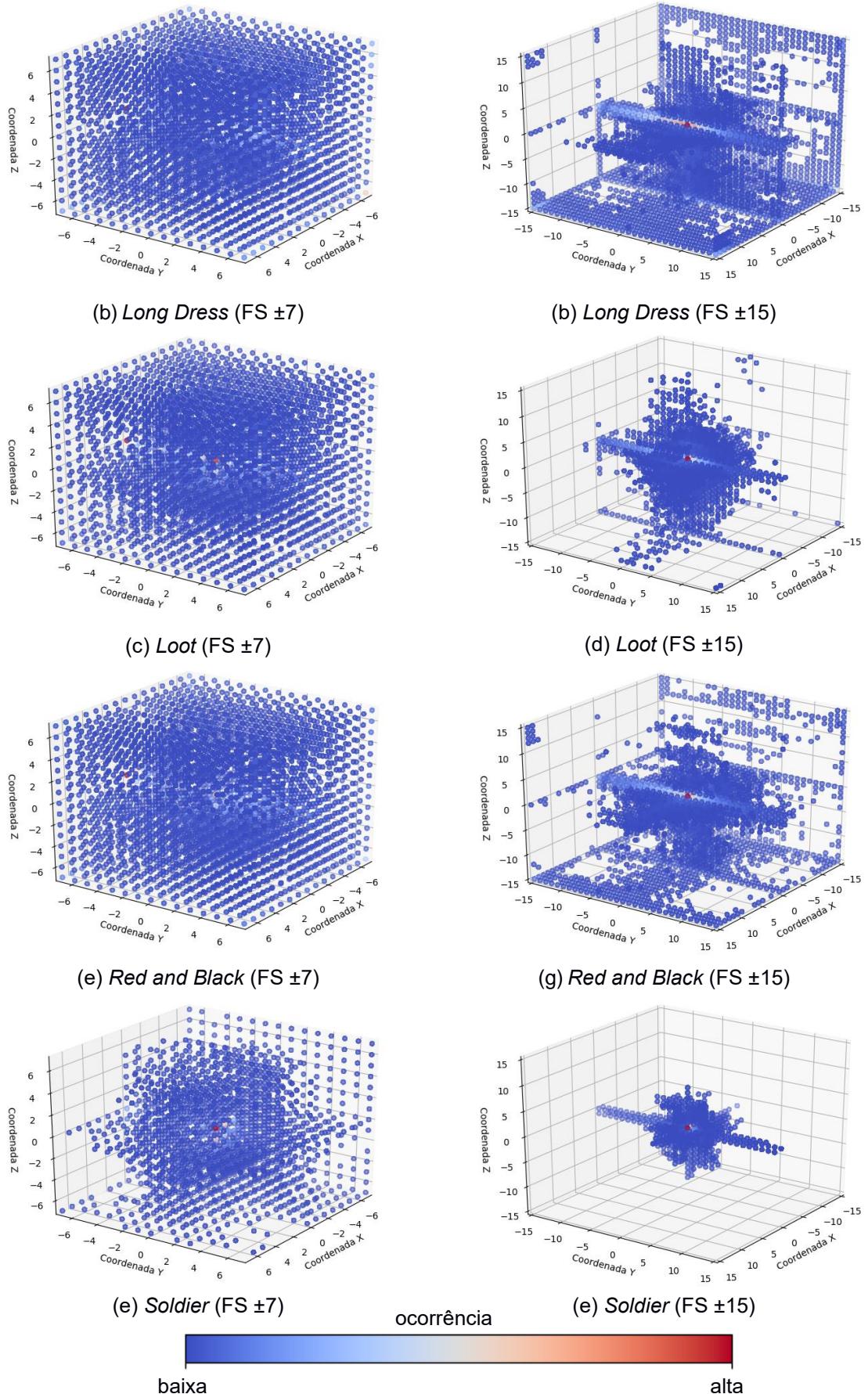


Figura 47: Mapas de calor para macroblocos de tamanho  $16 \times 16 \times 16$  voxels e algoritmo FS com deslocamento  $\pm 7$  e  $\pm 15$ .

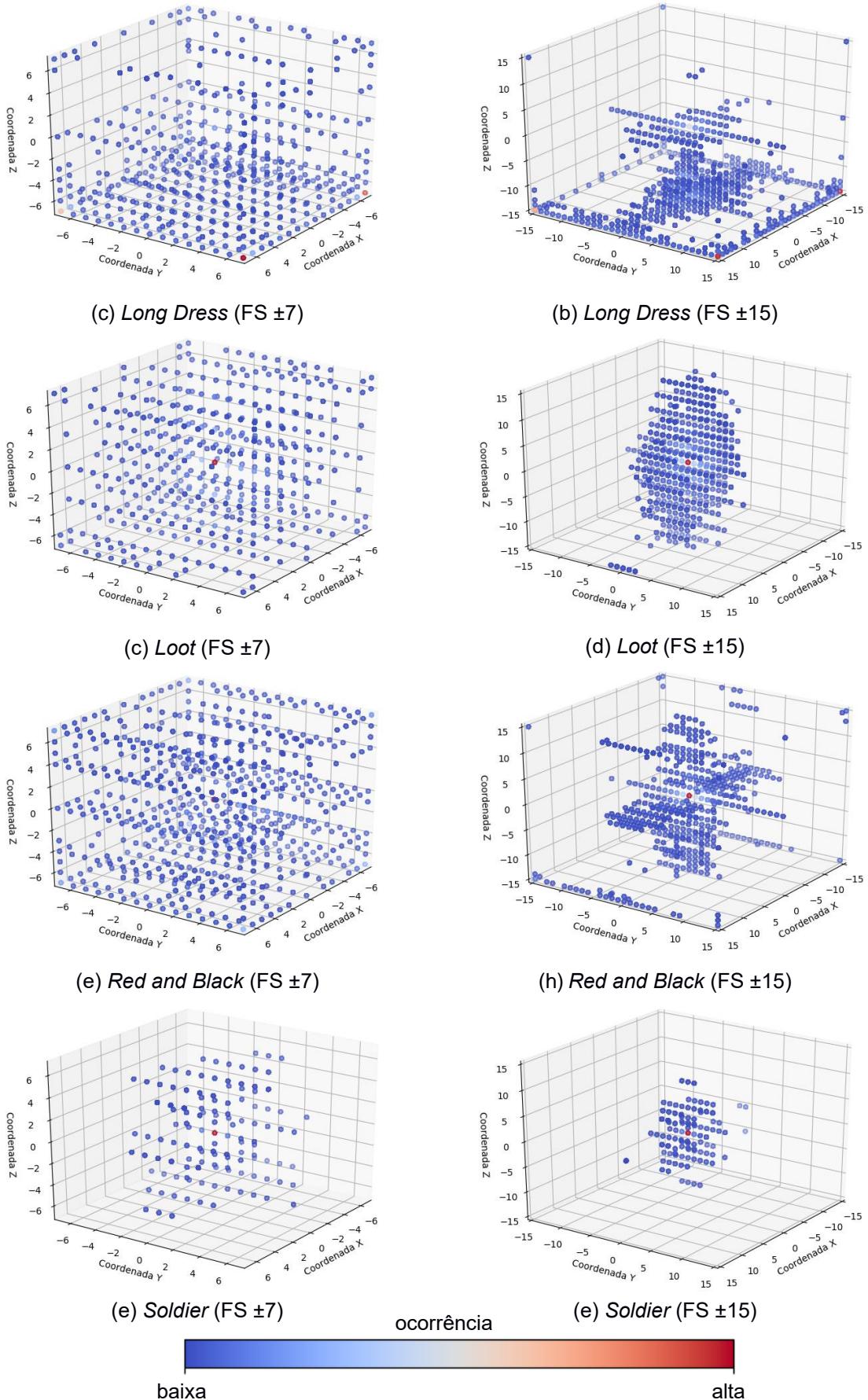


Figura 48: Mapas de calor para macroblocoes de tamanho  $32 \times 32 \times 32$  voxels e algoritmo FS com deslocamento  $\pm 7$  e  $\pm 15$  sem o algoritmo ICP.

Variações da solução desenvolvida também foram avaliadas com o objetivo de comprovar que a solução proposta com macroblocos de tamanho 32x32x32 (Proposta DS) e também a solução proposta com macroblocos de tamanho 16x16x16 (Proposta DS), apresenta maior eficiência entre outras possibilidades apresentadas. Neste sentido, a Figura 49 mostra um gráfico RD de geometria e também um gráfico RD de textura para os resultados médios de todas as sequências. Nesses são apresentados resultados de abordagens tais como: a solução proposta com uma ME que considera MSE tanto de geometria como de textura com mesmo peso (Solução geo-tex), e a solução proposta com *N-Step Search* (NSS), que nesse caso é 4SS (Solução NSS).

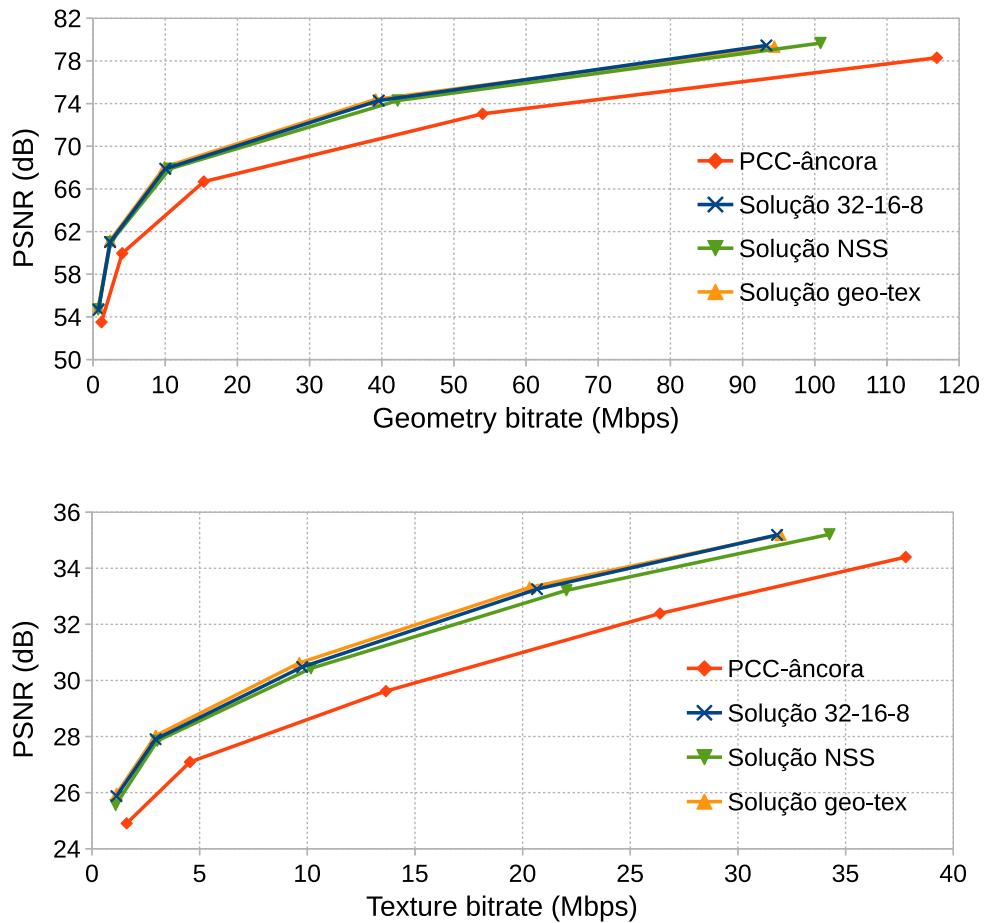


Figura 49: Resultados de variantes da abordagem proposta.

As figuras que seguem apresentam gráficos RD individuais para as sequências testadas.

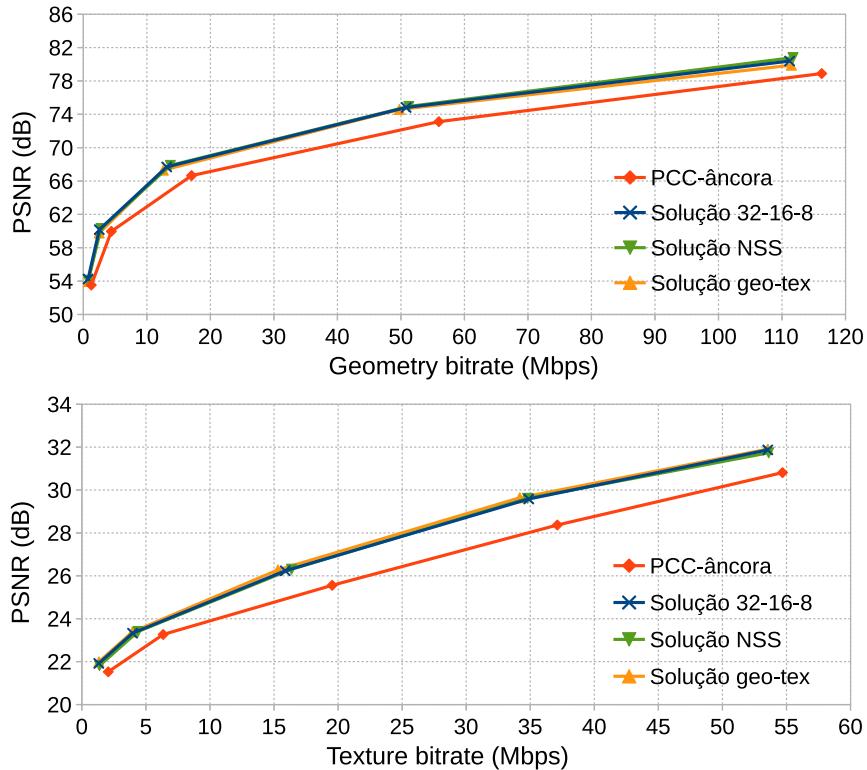


Figura 50: Resultados para sequência *Long Dress* de variações da proposta desenvolvida neste trabalho.

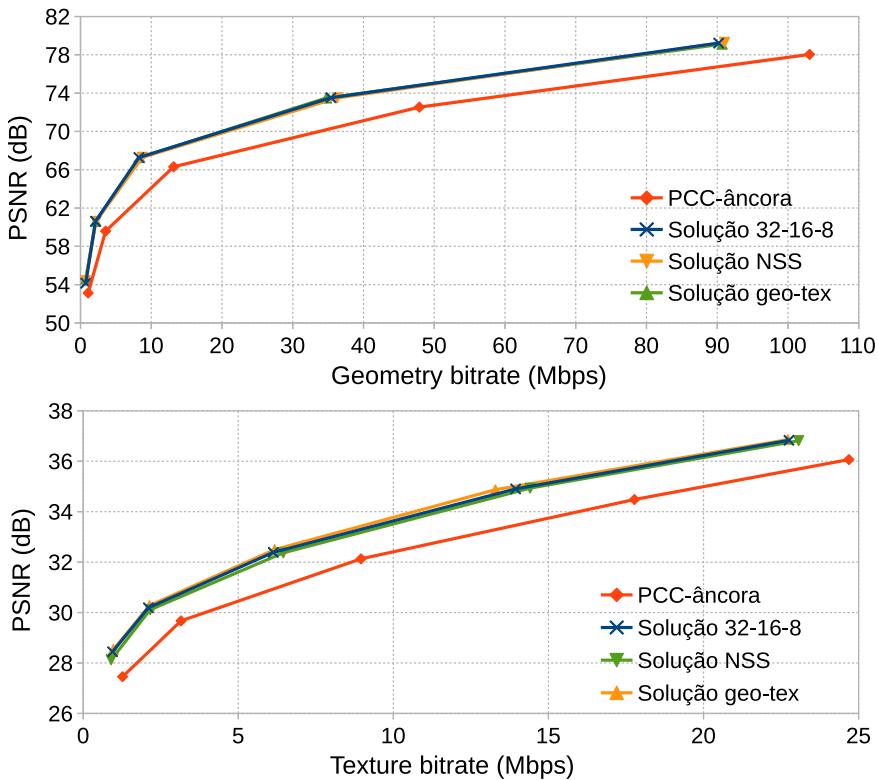


Figura 51: Resultados para sequência *Loot* de variações da proposta desenvolvida neste trabalho.

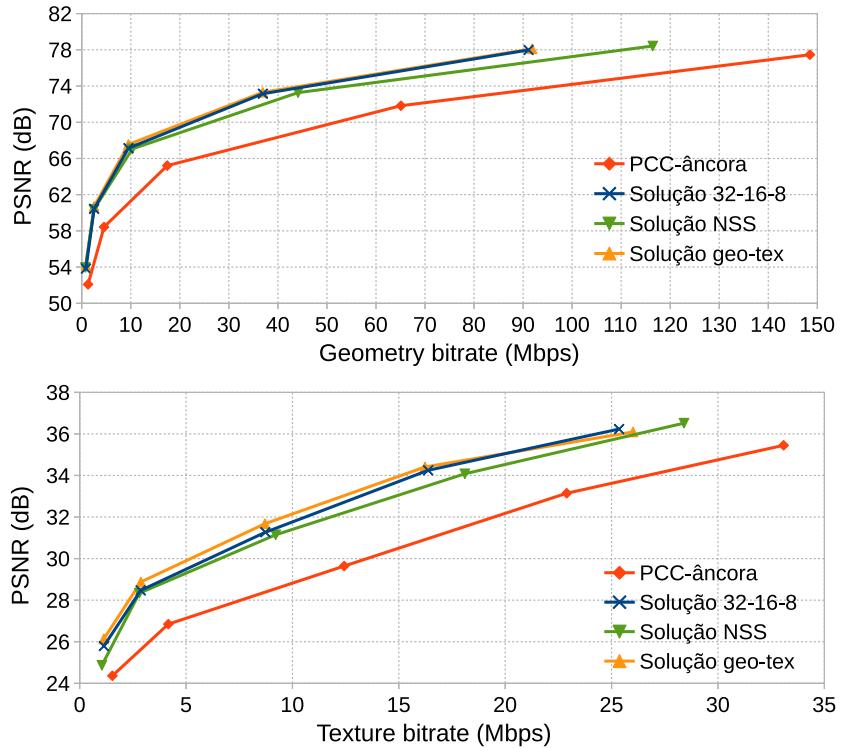


Figura 52: Resultados para sequência *Soldier* de variações da proposta desenvolvida neste trabalho.

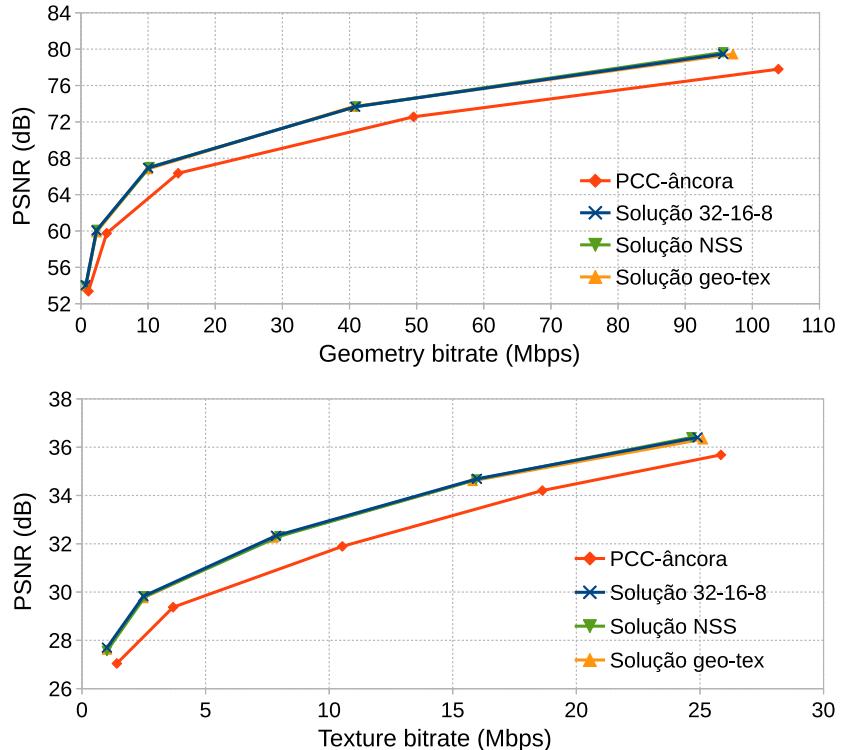


Figura 53: Resultados para sequência *Red and Black* de variações da proposta desenvolvida neste trabalho.

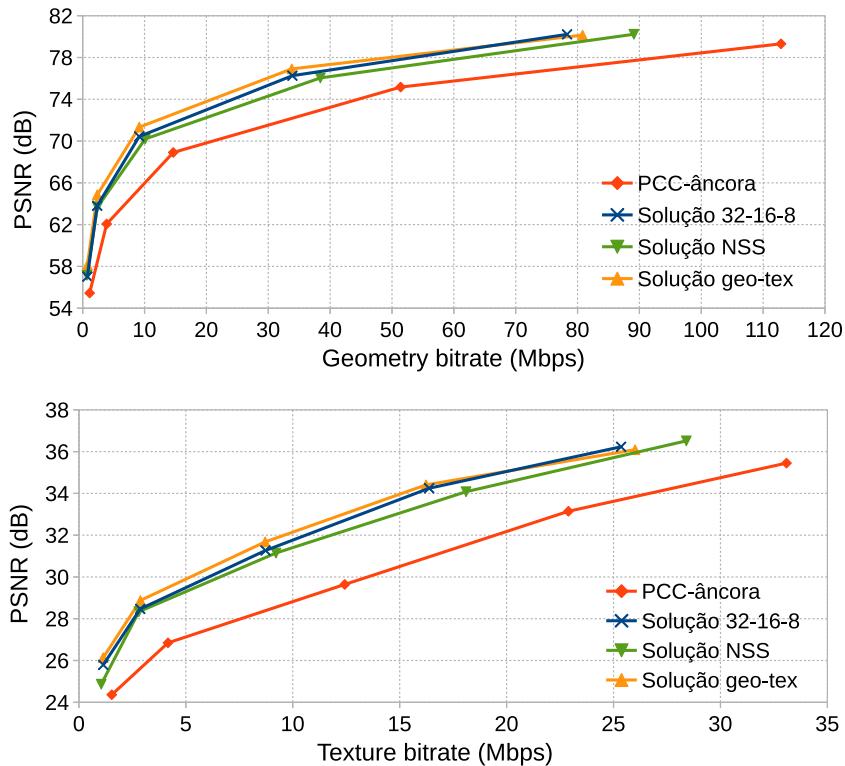


Figura 54: Resultados para sequência *Queen* de variações da proposta desenvolvida neste trabalho.

As Tabela XXIII e a Tabela XXIV apresentam resultados de BD-Rate e BD-PSNR da solução de subparticionamento e alinhamento das caixas delimitadoras publicada na *Visual Communications and Image Processing Conference 2018* (Santos et al., 2018) em comparação com a solução completa apresenta nesta tese.

Tabela XXIII: Resultados de *Bjontegaard-Delta* da abordagem desenvolvida, e o trabalho anterior em relação ao *PCC-âncora*.

Sequências	geo.BD-RATE (%)				geo.BD-PSNR (dB)			
	SUB(VCIP)		Solução 32		SUB(VCIP)		Solução 32	
	D1	D2	D1	D2	D1	D2	D1	D2
<i>Long Dress</i>	-13,29	-31,47	-25,47	-33,89	0,73	2,14	1,4	2,26
<i>Loot</i>	-18,55	-29,92	-38,35	-43,74	0,98	1,89	2,21	2,92
<i>Red and Black</i>	-13,49	-21,81	-3,5	-36,38	0,71	1,35	1,77	2,36
<i>Soldier</i>	-17,64	-34,72	-49,31	-57,7	0,91	2,25	3,06	4,35
<i>Queen</i>	-21,89	-36,50	-45,07	-53,94	1,16	2,27	2,71	3,75
<b>Média</b>	<b>-16,97</b>	<b>-30,88</b>	<b>-37,94</b>	<b>-45,13</b>	<b>0,90</b>	<b>1,98</b>	<b>2,23</b>	<b>3,13</b>

Tabela XXIV: Resultados de *Bjontegaard-Delta* da abordagem desenvolvida, e o trabalho anterior em relação ao PCC-âncora.

Sequências	Attr.BD-RATE (%)						Attr.BD-PSNR (dB)					
	SUB(VCIP)			Solução 32			SUB(VCIP)			Solução 32		
	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr	Luma	Cb	Cr
<i>Long</i>	-30.04	-5.16	-7.28	-30.53	-20.27	-20.74	1.28	0.08	0.12	1.26	0.3	0.33
<i>Loot</i>	-25.92	-11.00	-12.62	-35.64	-33.64	-35.95	0.91	0.24	0.29	1.3	0.75	0.84
<i>Red</i>	-23.21	-8.99	-14.84	-33.33	-30.7	-17.99	0.88	0.15	0.29	1.25	0.5	0.32
<i>Soldier</i>	-37.70	-14.70	-13.68	-56.43	-49.43	-49.47	1.58	0.21	0.21	2.65	0.87	0.94
<i>Queen</i>	-41.97	-16.66	-21.16	-51.83	-40.21	-41.26	2.34	0.38	0.55	3.02	1.05	1.2
<b>Média</b>	<b>-31.77</b>	<b>-11.30</b>	<b>-13.92</b>	<b>-41.55</b>	<b>-34.85</b>	<b>-33.08</b>	<b>1.40</b>	<b>0.21</b>	<b>0.29</b>	<b>1.90</b>	<b>0.69</b>	<b>0.72</b>

As figuras que seguem apresentam resultados completos para o esquema de alinhamento em relação ao PCC-âncora.

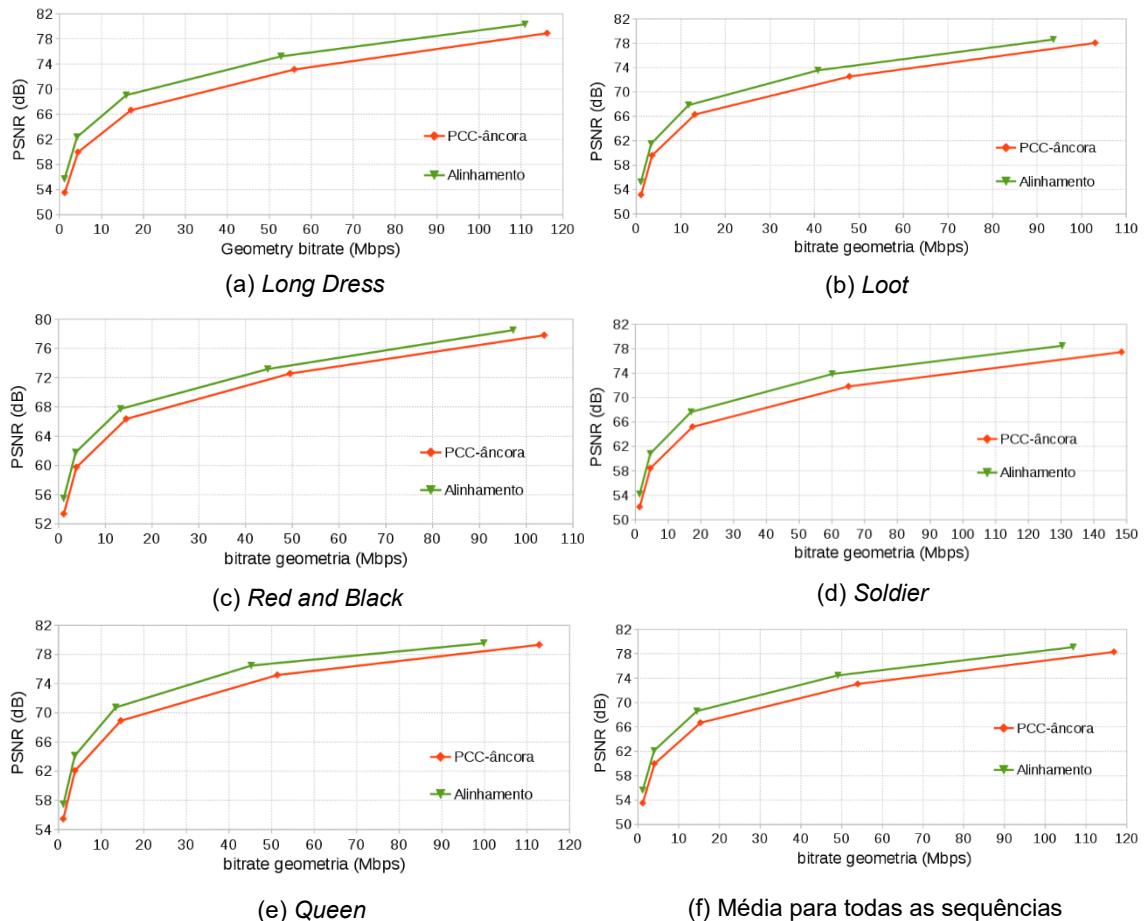


Figura 55: Gráficos RD de geometria para a proposta de alinhamento das caixas delimitadoras.

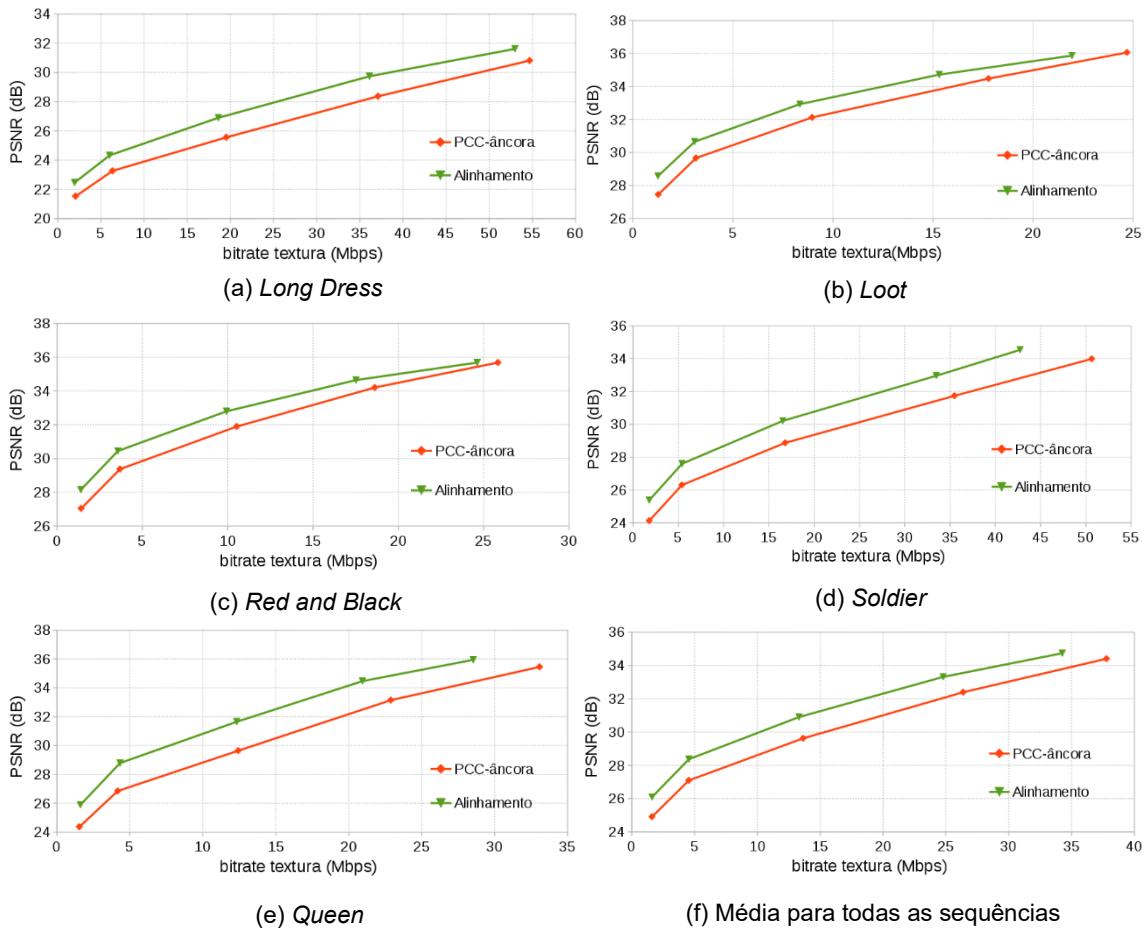


Figura 56: Gráficos RD para textura para a proposta de alinhamento das caixas delimitadoras.

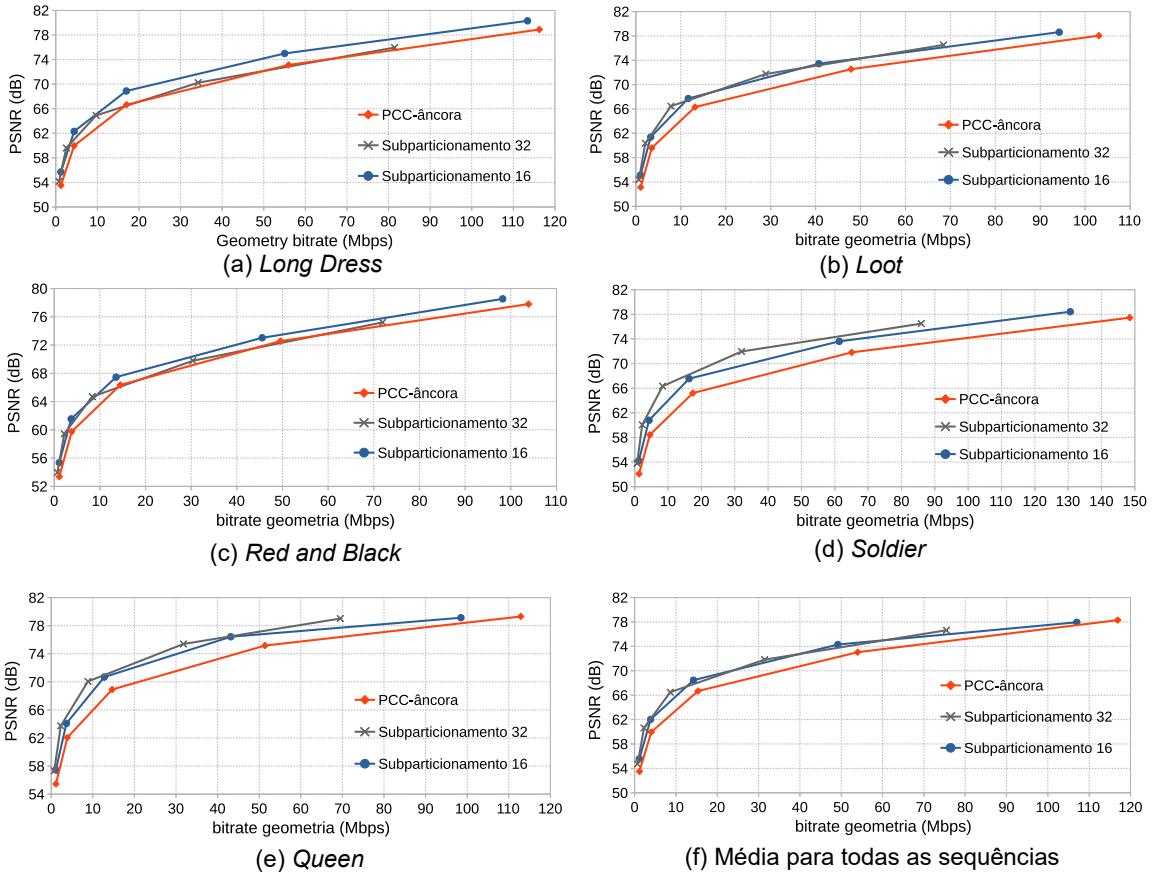


Figura 57: Gráficos RD para geometria para a proposta de Subparticionamento.

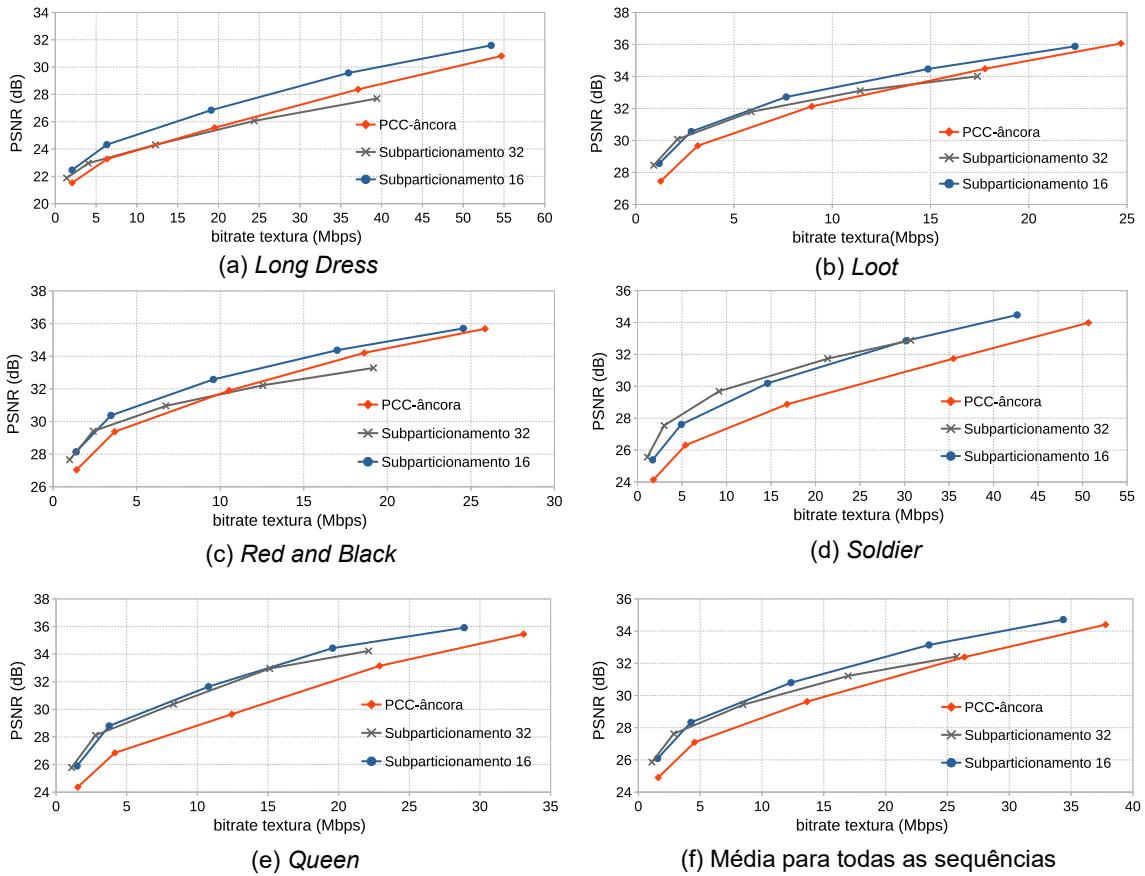


Figura 58: Gráficos RD para textura para a proposta de Subparticionamento.

Tabela XXV: Resultados para os dados de geometria do alinhamento das caixas delimitadoras em comparação ao PCC-âncora.

LoD	Sequências	PSNR (dB)		Taxa (Mbits/s)	
		PCC	Alinhamento	PCC	Alinhamento
g6	<i>Long</i>	53,52	55,70	1,25	1,24
	<i>Loot</i>	53,13	55,24	1,08	1,04
	<i>Red</i>	53,37	55,48	1,12	1,11
	<i>Soldier</i>	52,09	54,20	1,29	1,27
	<i>Queen</i>	55,45	57,45	1,13	1,15
	<b>Média</b>	<b>53,51</b>	<b>55,61</b>	<b>1,17</b>	<b>1,16</b>
g7	<i>Long</i>	59,96	62,36	4,41	4,39
	<i>Loot</i>	59,59	61,51	13,16	3,33
	<i>Red</i>	59,75	61,78	14,48	3,65
	<i>Soldier</i>	58,43	60,82	17,43	4,52
	<i>Queen</i>	62,06	64,11	14,64	3,82
	<b>Média</b>	<b>59,96</b>	<b>62,12</b>	<b>12,83</b>	<b>3,94</b>
g8	<i>Long</i>	66,65	69,04	17,04	16,85
	<i>Loot</i>	66,31	67,85	47,89	11,74
	<i>Red</i>	66,35	67,74	49,54	13,29
	<i>Soldier</i>	65,21	67,65	65,08	16,99
	<i>Queen</i>	68,91	70,73	51,39	13,44
	<b>Média</b>	<b>66,69</b>	<b>68,60</b>	<b>46,19</b>	<b>14,46</b>
g9	<i>Long</i>	73,12	75,21	55,97	54,53
	<i>Loot</i>	72,54	73,55	47,89	40,81
	<i>Red</i>	72,56	73,17	49,54	44,73
	<i>Soldier</i>	71,82	73,87	65,08	60,12
	<i>Queen</i>	75,17	76,47	51,39	45,25
	<b>Média</b>	<b>73,04</b>	<b>74,45</b>	<b>53,98</b>	<b>49,09</b>
g10	<i>Long</i>	78,89	80,31	116,25	113,26
	<i>Loot</i>	78,04	78,60	103,05	93,66
	<i>Red</i>	77,80	78,51	103,87	97,21
	<i>Soldier</i>	77,46	78,49	148,49	130,37
	<i>Queen</i>	79,31	79,54	112,88	99,86
	<b>Média</b>	<b>78,30</b>	<b>79,09</b>	<b>116,91</b>	<b>106,87</b>

Tabela XXVI: Resultados para os dados de textura do alinhamento das caixas delimitadoras em comparação ao PCC-âncora.

LoD	Sequências	PSNR (dB)		Taxa (Mbits/s)	
		PCC	Alinhamento	PCC	Alinhamento
g6	<i>Long</i>	21,53	22,46	2,06	2,06
	<i>Loot</i>	27,46	28,58	1,27	1,26
	<i>Red</i>	27,04	28,15	1,41	1,39
	<i>Soldier</i>	24,14	25,38	1,79	1,79
	<i>Queen</i>	24,36	25,89	1,54	1,61
	<b>Média</b>	<b>24,90</b>	<b>26,09</b>	<b>1,61</b>	<b>1,62</b>
g7	<i>Long</i>	23,26	24,33	6,35	6,38
	<i>Loot</i>	29,67	30,66	3,15	3,09
	<i>Red</i>	29,37	30,45	3,69	3,56
	<i>Soldier</i>	26,30	27,60	5,39	5,41
	<i>Queen</i>	26,85	28,78	4,17	4,32
	<b>Média</b>	<b>27,09</b>	<b>28,36</b>	<b>4,55</b>	<b>4,55</b>
g8	<i>Long</i>	25,56	26,90	19,54	19,46
	<i>Loot</i>	32,13	32,94	8,96	8,35
	<i>Red</i>	31,89	32,80	10,53	9,93
	<i>Soldier</i>	28,87	30,21	16,80	16,54
	<i>Queen</i>	29,64	31,65	12,43	12,33
	<b>Média</b>	<b>29,62</b>	<b>30,90</b>	<b>13,65</b>	<b>13,32</b>
g9	<i>Long</i>	28,37	29,74	37,12	36,67
	<i>Loot</i>	34,48	34,73	17,77	15,31
	<i>Red</i>	34,20	34,65	18,63	17,52
	<i>Soldier</i>	31,73	32,95	35,49	33,48
	<i>Queen</i>	33,14	34,47	22,89	20,94
	<b>Média</b>	<b>32,39</b>	<b>33,31</b>	<b>26,38</b>	<b>24,79</b>
g10	<i>Long</i>	30,81	31,62	54,69	53,50
	<i>Loot</i>	36,07	35,88	24,69	21,94
	<i>Red</i>	35,68	35,68	25,85	24,63
	<i>Soldier</i>	33,98	34,53	50,66	42,74
	<i>Queen</i>	35,45	35,94	33,09	28,54
	<b>Média</b>	<b>34,40</b>	<b>34,73</b>	<b>37,79</b>	<b>34,27</b>

Tabela XXVII: Resultados para os dados de geometria do subparticionamento de macroblocos em comparação ao PCC-âncora.

LoD	Sequências	PSNR(dB)			Taxa (Mbits/s)		
		PCC	SUB 16	SUB 32	PCC	SUB 16	SUB 32
g6	<i>Long</i>	53,52	55,70	54,15	1,25	1,24	0,78
	<i>Loot</i>	53,13	55,13	54,36	1,08	1,03	0,68
	<i>Red</i>	53,37	55,37	53,97	1,12	1,12	0,71
	<i>Soldier</i>	52,09	54,19	53,80	1,29	1,21	0,71
	<i>Queen</i>	55,45	57,44	57,32	1,13	1,11	0,72
	<b>Média</b>	<b>53,51</b>	<b>55,57</b>	<b>54,72</b>	<b>1,17</b>	<b>1,14</b>	<b>0,72</b>
g7	<i>Long</i>	59,96	62,31	59,59	4,41	4,42	2,53
	<i>Loot</i>	59,59	61,38	60,36	13,16	3,31	2,11
	<i>Red</i>	59,75	61,54	59,44	14,48	3,71	2,21
	<i>Soldier</i>	58,43	60,80	60,04	17,43	4,31	2,24
	<i>Queen</i>	62,06	64,08	63,73	14,64	3,66	2,30
	<b>Média</b>	<b>59,96</b>	<b>62,02</b>	<b>60,63</b>	<b>12,83</b>	<b>3,88</b>	<b>2,28</b>
g8	<i>Long</i>	66,65	68,88	64,91	17,04	16,96	9,72
	<i>Loot</i>	66,31	67,71	66,45	47,89	11,66	7,81
	<i>Red</i>	66,35	67,47	64,70	49,54	13,55	8,35
	<i>Soldier</i>	65,21	67,55	66,33	65,08	16,30	8,37
	<i>Queen</i>	68,91	70,68	70,06	51,39	12,75	8,83
	<b>Média</b>	<b>66,69</b>	<b>68,46</b>	<b>66,49</b>	<b>46,19</b>	<b>14,25</b>	<b>8,62</b>
g9	<i>Long</i>	73,12	74,99	70,23	55,97	55,03	34,21
	<i>Loot</i>	72,54	73,43	71,74	47,89	40,73	28,93
	<i>Red</i>	72,56	73,03	69,78	49,54	45,55	30,37
	<i>Soldier</i>	71,82	73,61	71,96	65,08	61,34	32,04
	<i>Queen</i>	75,17	76,42	75,39	51,39	43,16	31,77
	<b>Média</b>	<b>73,04</b>	<b>74,30</b>	<b>71,82</b>	<b>53,98</b>	<b>49,17</b>	<b>31,47</b>
g10	<i>Long</i>	78,89	80,30	75,95	116,25	113,41	81,45
	<i>Loot</i>	78,04	78,62	76,51	103,05	94,22	68,47
	<i>Red</i>	77,80	78,54	75,21	103,87	98,23	71,93
	<i>Soldier</i>	77,46	78,42	76,49	148,49	130,62	85,93
	<i>Queen</i>	79,31	79,13	79,02	112,88	98,49	69,50
	<b>Média</b>	<b>78,30</b>	<b>79,00</b>	<b>76,64</b>	<b>116,91</b>	<b>106,99</b>	<b>75,46</b>

Tabela XXVIII: Resultados para os dados de textura do subparticionamento de macroblocos em comparação ao PCC-âncora.

LoD	Sequências	PSNR(dB)			Taxa (Mbits)		
		PCC	SUB 16	SUB 32	PCC	SUB 16	SUB 32
g6	<i>Long</i>	21,53	22,46	21,87	2,06	2,05	1,39
	<i>Loot</i>	27,46	28,56	28,45	1,27	1,19	0,91
	<i>Red</i>	27,04	28,14	27,66	1,41	1,37	0,99
	<i>Soldier</i>	24,14	25,38	25,56	1,79	1,69	1,10
	<i>Queen</i>	24,36	25,89	25,78	1,54	1,48	1,10
	<b>Média</b>	<b>24,90</b>	<b>26,09</b>	<b>25,86</b>	<b>1,61</b>	<b>1,56</b>	<b>1,10</b>
g7	<i>Long</i>	23,26	24,32	22,96	6,35	6,32	4,04
	<i>Loot</i>	29,67	30,56	30,07	3,15	2,82	2,12
	<i>Red</i>	29,37	30,37	29,41	3,69	3,46	2,40
	<i>Soldier</i>	26,30	27,61	27,54	5,39	4,92	2,99
	<i>Queen</i>	26,85	28,79	28,11	4,17	3,76	2,79
	<b>Média</b>	<b>27,09</b>	<b>28,33</b>	<b>27,62</b>	<b>4,55</b>	<b>4,26</b>	<b>2,87</b>
g8	<i>Long</i>	25,56	26,85	24,31	19,54	19,11	12,27
	<i>Loot</i>	32,13	32,72	31,79	8,96	7,66	5,87
	<i>Red</i>	31,89	32,58	30,96	10,53	9,59	6,75
	<i>Soldier</i>	28,87	30,19	29,69	16,80	14,61	9,15
	<i>Queen</i>	29,64	31,64	30,38	12,43	10,79	8,31
	<b>Média</b>	<b>29,62</b>	<b>30,79</b>	<b>29,42</b>	<b>13,65</b>	<b>12,35</b>	<b>8,47</b>
g9	<i>Long</i>	28,37	29,57	26,06	37,12	35,94	24,37
	<i>Loot</i>	34,48	34,47	33,11	17,77	14,87	11,42
	<i>Red</i>	34,20	34,37	32,22	18,63	17,00	12,55
	<i>Soldier</i>	31,73	32,85	31,73	35,49	30,17	21,36
	<i>Queen</i>	33,14	34,43	32,95	22,89	19,57	15,12
	<b>Média</b>	<b>32,39</b>	<b>33,14</b>	<b>31,21</b>	<b>26,38</b>	<b>23,51</b>	<b>16,97</b>
g10	<i>Long</i>	30,81	31,59	27,70	54,69	53,43	39,42
	<i>Loot</i>	36,07	35,88	34,01	24,69	22,35	17,39
	<i>Red</i>	35,68	35,70	33,28	25,85	24,54	19,16
	<i>Soldier</i>	33,98	34,47	32,88	50,66	42,64	30,69
	<i>Queen</i>	35,45	35,91	34,23	33,09	28,88	22,12
	<b>Média</b>	<b>34,40</b>	<b>34,71</b>	<b>32,42</b>	<b>37,79</b>	<b>34,37</b>	<b>25,76</b>

Tabela XXIX: Resultados comparativo de qualidade de geometria e taxa de bits para a solução proposta iniciando com macroblocos 16x16x16, a solução iniciando com macroblocos 32x32x32 e o PCC-âncora.

LoD	Sequências	PSNR(dB)			Taxa (Mbits/s)		
		PCC	Solução 16	Solução 32	PCC	Solução 16	Solução 32
g6	<b>Long</b>	53,52	54,41	54,26	1,25	0,83	0,75
	<b>Loot</b>	53,13	54,92	54,15	1,08	0,78	0,74
	<b>Red</b>	53,37	54,50	54,04	1,12	0,80	0,72
	<b>Soldier</b>	52,09	54,43	53,83	1,29	0,87	0,85
	<b>Queen</b>	55,45	58,43	57,02	1,13	0,82	0,76
	<b>Média</b>	<b>53,51</b>	<b>55,34</b>	<b>54,66</b>	<b>1,17</b>	<b>0,82</b>	<b>0,76</b>
g7	<b>Long</b>	59,96	61,01	60,17	4,41	3,04	2,51
	<b>Loot</b>	59,59	61,33	60,62	13,16	2,53	2,12
	<b>Red</b>	59,75	60,90	60,06	14,48	2,72	2,32
	<b>Soldier</b>	58,43	61,04	60,43	17,43	2,94	2,52
	<b>Queen</b>	62,06	65,07	63,80	14,64	2,71	2,35
	<b>Média</b>	<b>59,96</b>	<b>61,87</b>	<b>61,02</b>	<b>12,83</b>	<b>2,79</b>	<b>2,37</b>
g8	<b>Long</b>	66,65	68,36	67,69	17,04	13,87	13,12
	<b>Loot</b>	66,31	68,02	67,28	47,89	10,18	8,26
	<b>Red</b>	66,35	67,88	66,94	49,54	11,37	10,00
	<b>Soldier</b>	65,21	67,69	67,13	65,08	11,47	9,52
	<b>Queen</b>	68,91	71,54	70,44	51,39	10,85	9,14
	<b>Média</b>	<b>66,69</b>	<b>68,70</b>	<b>67,90</b>	<b>46,19</b>	<b>11,55</b>	<b>10,01</b>
g9	<b>Long</b>	73,12	75,10	74,82	55,97	51,21	50,82
	<b>Loot</b>	72,54	74,16	73,52	47,89	40,56	35,40
	<b>Red</b>	72,56	74,28	73,66	49,54	43,50	40,85
	<b>Soldier</b>	71,82	73,64	73,15	65,08	45,08	36,93
	<b>Queen</b>	75,17	77,24	76,26	51,39	40,80	33,90
	<b>Média</b>	<b>73,04</b>	<b>74,88</b>	<b>74,28</b>	<b>53,98</b>	<b>44,23</b>	<b>39,58</b>
g10	<b>Long</b>	78,89	80,95	80,39	116,25	111,95	111,19
	<b>Loot</b>	78,04	79,81	79,22	103,05	100,07	90,20
	<b>Red</b>	77,80	80,07	79,47	103,87	102,02	95,66
	<b>Soldier</b>	77,46	78,58	77,98	148,49	120,26	91,12
	<b>Queen</b>	79,31	80,65	78,01	112,88	104,63	78,30
	<b>Média</b>	<b>78,30</b>	<b>80,01</b>	<b>79,01</b>	<b>116,91</b>	<b>107,79</b>	<b>93,29</b>

Tabela XXX: Resultados comparativo de qualidade de textura e taxa de bits para a solução proposta iniciando com macroblocos 16x16x16, a solução iniciando com macroblocos 32x32x32 e o PCC-âncora.

LoD	Sequências	PSNR(dB)			Taxa (Mbits/s)		
		PCC	Solução 16	Solução 32	PCC	Solução 16	Solução 32
g6	Long	21,53	21,97	21,92	2,06	1,35	1,32
	Loot	27,46	28,55	28,45	1,27	0,94	0,95
	Red	27,04	27,75	27,70	1,41	1,02	1,01
	Soldier	24,14	25,59	25,52	1,79	1,20	1,28
	Queen	24,36	26,21	25,79	1,54	1,12	1,14
	Média	<b>24,90</b>	<b>26,01</b>	<b>25,87</b>	<b>1,61</b>	<b>1,13</b>	<b>1,14</b>
g7	Long	23,26	23,59	23,33	6,35	4,33	3,96
	Loot	29,67	30,34	30,19	3,15	2,20	2,10
	Red	29,37	29,89	29,83	3,69	2,60	2,49
	Soldier	26,30	27,88	27,71	5,39	3,38	3,35
	Queen	26,85	28,92	28,47	4,17	2,86	2,87
	Média	<b>27,09</b>	<b>28,12</b>	<b>27,90</b>	<b>4,55</b>	<b>3,07</b>	<b>2,95</b>
g8	Long	25,56	26,45	26,24	19,54	16,05	15,87
	Loot	32,13	32,55	32,39	8,96	6,63	6,13
	Red	31,89	32,29	32,33	10,53	8,07	7,85
	Soldier	28,87	30,41	30,17	16,80	10,34	10,24
	Queen	29,64	31,78	31,27	12,43	8,76	8,73
	Média	<b>29,62</b>	<b>30,69</b>	<b>30,48</b>	<b>13,65</b>	<b>9,97</b>	<b>9,76</b>
g9	Long	28,37	29,67	29,58	37,12	34,43	34,88
	Loot	34,48	35,01	34,90	17,77	14,35	13,94
	Red	34,20	34,61	34,69	18,63	16,06	15,98
	Soldier	31,73	33,07	32,85	35,49	22,74	22,16
	Queen	33,14	34,49	34,25	22,89	16,69	16,38
	Média	<b>32,39</b>	<b>33,37</b>	<b>33,25</b>	<b>26,38</b>	<b>20,85</b>	<b>20,67</b>
g10	Long	30,81	31,92	31,87	54,69	52,12	53,54
	Loot	36,07	36,85	36,83	24,69	22,96	22,75
	Red	35,68	36,28	36,40	25,85	24,58	24,90
	Soldier	33,98	34,71	34,61	50,66	34,50	32,53
	Queen	35,45	36,12	36,23	33,09	26,63	25,35
	Média	<b>34,40</b>	<b>35,18</b>	<b>35,19</b>	<b>37,79</b>	<b>32,16</b>	<b>31,82</b>

## APÊNDICE B – RESULTADOS DA RECONSTRUÇÃO DAS NUVENS DE PONTOS

Este apêndice apresenta os resultados visuais através de nuvens de pontos reconstruídas por meio de algumas explorações realizadas no decorrer da elaboração dessa tese, assim como, o resultado final da solução proposta.

A figuras que seguem apresentam nuvens de pontos reconstruídas com a solução proposta iniciando com marcoblocos de tamanho 32 em comparação com a ao PCC-âncora e a também a nuvem de pontos original sem codificação.



Figura 59: Resultados da solução proposta para a nuvem de número 1052 da sequência *Long Dress*.



Figura 60: Resultados da solução proposta para a nuvem de número 1001 da sequência *Loot*.



Figura 61: Resultados da solução proposta para a nuvem de número 1451 da sequência *Red and Black*.

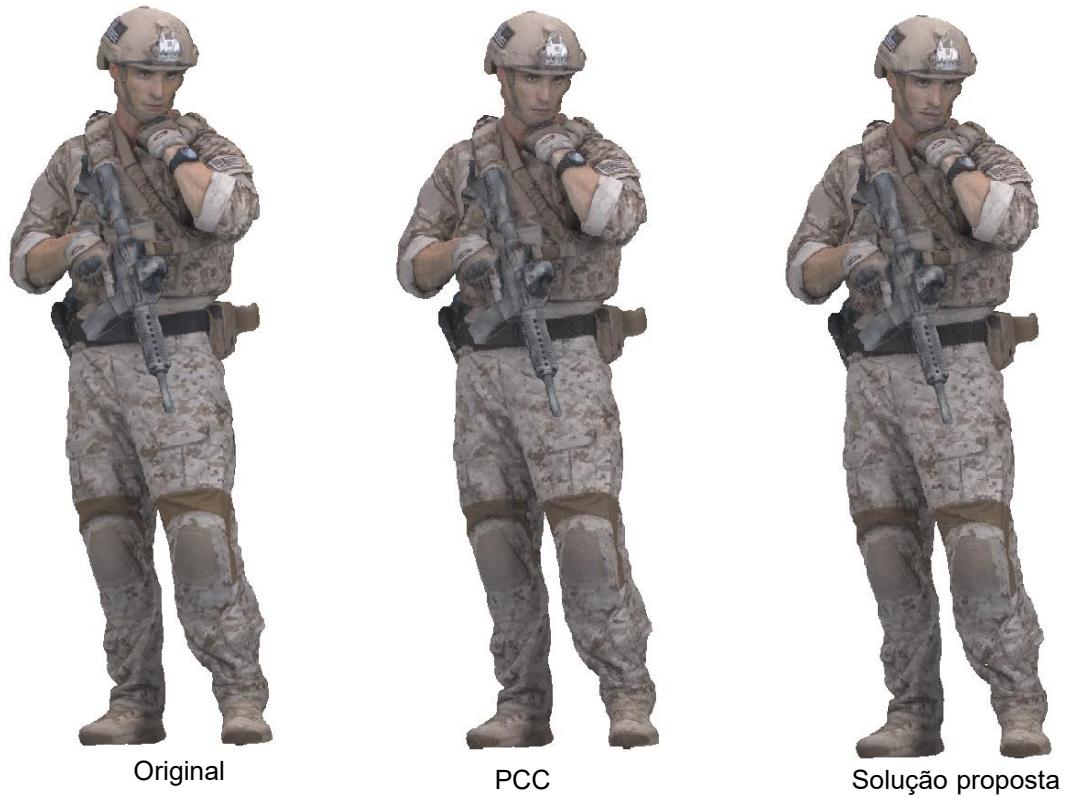


Figura 62: Resultados da solução proposta para a nuvem de número 537 da sequência *Soldier*.

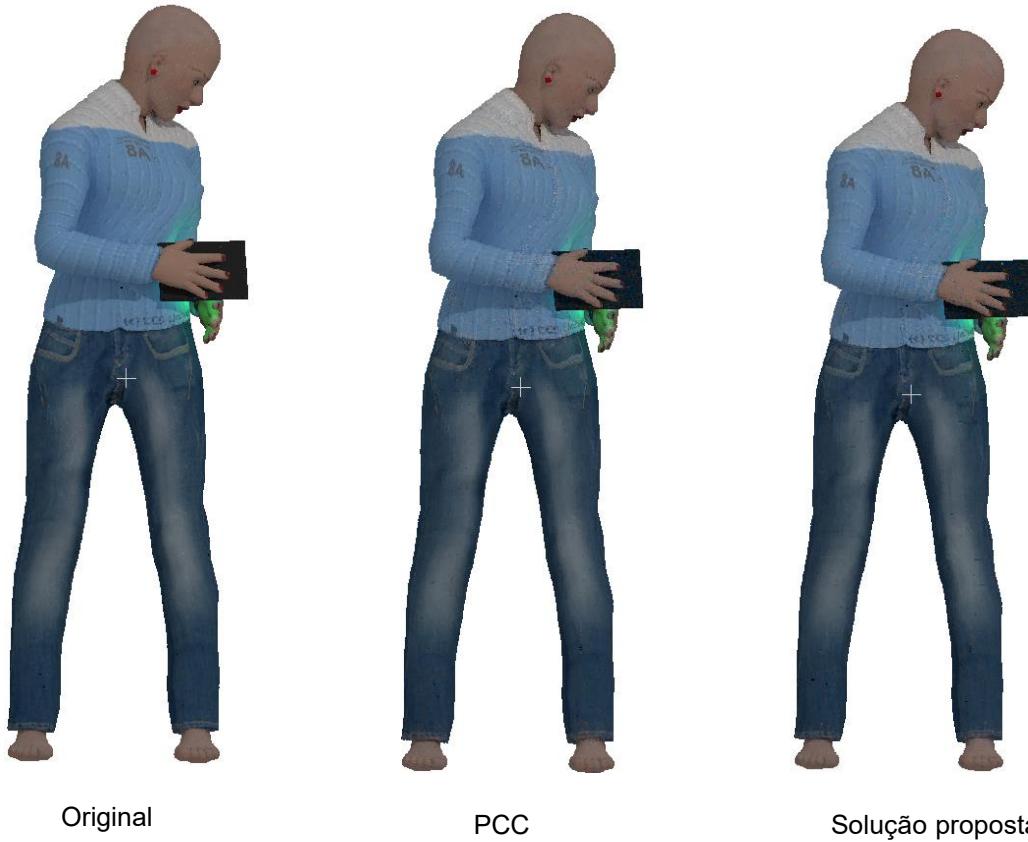


Figura 63: Resultados da solução proposta para a nuvem de número 0001 da sequência *Queen*.

As figuras que seguem são nuvens de pontos reconstruídas para alguns experimentos realizados para definição da decisão de modo.



Figura 64: Resultados das decisões de modo testadas para *Long Dress 1052*.

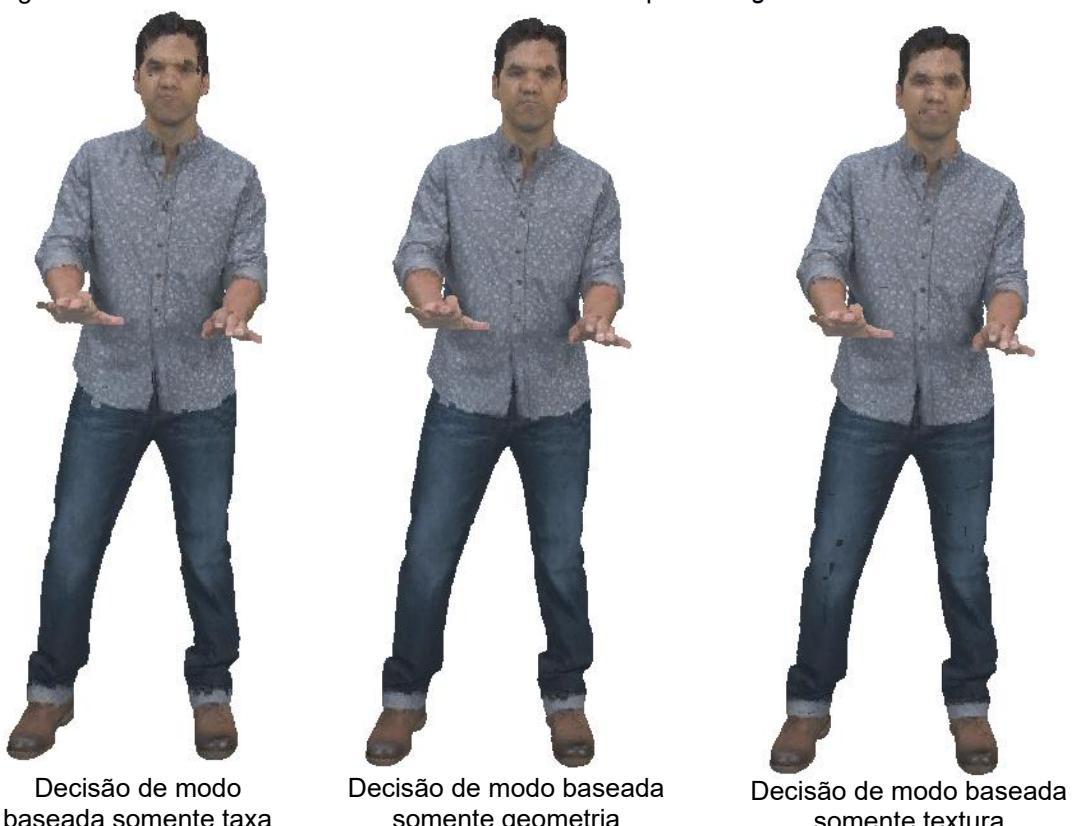


Figura 65: Resultados das decisões de modo testadas para *Loot 1001*.

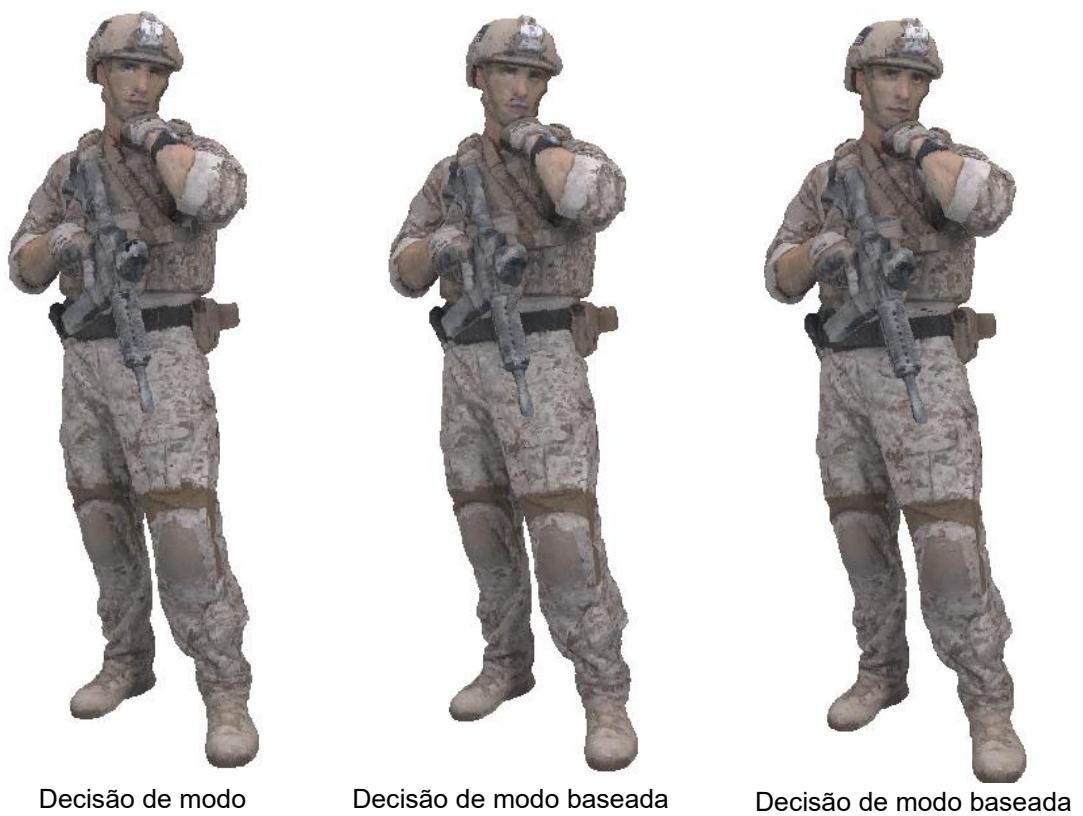


Decisão de modo baseada somente taxa

Decisão de modo baseada somente geometria

Decisão de modo baseada somente textura

Figura 66: Resultados das decisões de modo testadas para *Red and Black 1451*



Decisão de modo baseada somente taxa

Decisão de modo baseada somente geometria

Decisão de modo baseada somente textura

Figura 67 Resultados das decisões de modo testadas para *Soldier 537*.

No QR code abaixo é possível conferir imagens de nuvens de pontos reconstruídas de experimentos realizados com variações da decisão de modo. Além disso, também são disponibilizados pequenos segmentos de vídeos que mostram os ganhos em número de pontos codificados pela predição inter-nuvens. Por fim, algumas imagens de nuvens de pontos reconstruídas pela solução proposta são colocadas lado a lado com as nuvens reconstruídas pelo PCC-âncora;



Figura 68: QR CODE para apresentação de resultados visuais adicionais dos passos do desenvolvimento.

URL para a página web: <https://tinyurl.com/sI6czav>