

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



Dissertação

**Redução de Custo Computacional da Codificação de Nuvens de Pontos
Dinâmicas**

Gustavo Hübner Rehbein

Pelotas, 2025

Gustavo Hübner Rehbein

**Redução de Custo Computacional da Codificação de Nuvens de Pontos
Dinâmicas**

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

Orientador: Prof. Dr. Marcelo Schiavon Porto
Coorientadores: Prof. Dr. Guilherme Ribeiro Corrêa
Prof. Dr. Cristiano Flores dos Santos

Pelotas, 2025

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

R345r Rehbein, Gustavo Hübner

Redução de custo computacional da codificação de nuvens de pontos dinâmicas [recurso eletrônico] / Gustavo Hübner Rehbein ; Marcelo Schiavon Porto, orientador ; Guilherme Ribeiro Corrêa, Cristiano Flores dos Santos, coorientadores. — Pelotas, 2025.
90 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2025.

1. Compressão de nuvens de pontos. 2. Video-based point cloud compression;. 3. Aprendizado de máquina. 4. Redução de custo computacional. I. Porto, Marcelo Schiavon, orient. II. Corrêa, Guilherme Ribeiro, coorient. III. Santos, Cristiano Flores dos, coorient. IV. Título.

CDD 005

Gustavo Hübner Rehbein

**Redução de Custo Computacional da Codificação de Nuvens de Pontos
Dinâmicas**

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

Data da Defesa: 18 de julho de 2025

Banca Examinadora:

Prof. Dr. Marcelo Schiavon Porto (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Luciano Volcan Agostini

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Marilton Sanchotene de Aguiar

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Ismael Seidel

Doutor em Computação pela Universidade Federal de Santa Catarina.

AGRADECIMENTOS

Gostaria de agradecer e dedicar este trabalho aos meus pais, Denise Maria Rehbein e Soni Rehbein, por todo o amor, apoio e incentivo incondicionais.

Em especial, dedico este trabalho ao meu pai, que durante o período de desenvolvimento desta dissertação enfrentou o desafio de um diagnóstico de câncer, mas, com muita força e determinação, conseguiu superar essa fase difícil. Sua coragem e perseverança foram fonte de inspiração para mim ao longo desta jornada.

Estendo meus agradecimentos ao meu orientador, Marcelo Porto, pela orientação, paciência e dedicação ao longo desta jornada, bem como aos coorientadores, Guilherme Corrêa e Cristiano Santos, que contribuíram de maneira significativa para o desenvolvimento deste trabalho. Sou grato também ao aluno de iniciação científica Eduardo Costa, cuja colaboração foi essencial para o desenvolvimento de parte dos experimentos e análises apresentados nesta dissertação.

Agradeço ainda aos demais companheiros e amigos do grupo de pesquisas ViTech, pelo apoio, troca de conhecimentos e amizade, que foram fundamentais para o sucesso desta dissertação.

RESUMO

REHBEIN, Gustavo Hübner. **Redução de Custo Computacional da Codificação de Nuvens de Pontos Dinâmicas**. Orientador: Marcelo Schiavon Porto. 2025. 90 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

Nuvens de pontos são um formato de mídia digital tridimensional amplamente utilizado em diversas aplicações, como mapeamento de objetos históricos, vídeos 3D tele-imersivos, visualização em realidade virtual ou aumentada e navegação autônoma de veículos. Nuvens de pontos dinâmicas, que representam uma sequência de nuvens de pontos, são análogas a vídeos 2D digitais e apresentam um grande volume de dados. O *Video-based Point Cloud Compression* (V-PCC) é um padrão de compressão de nuvens de pontos dinâmicas criado pelo MPEG, que utiliza codificadores de vídeo 2D convencionais para comprimir os dados. No entanto, a etapa de codificação de vídeo do *software* de referência do V-PCC representa cerca de 90% do custo computacional total do processo de codificação, tornando o V-PCC desafiador para aplicações em tempo real e em dispositivos com restrições de energia. Esta dissertação apresenta uma solução para a redução do custo computacional do V-PCC, focada na etapa de codificação de vídeo dos subfluxos do V-PCC. Modelos de aprendizado de máquina baseados em árvores de decisão são utilizados para realizar a interrupção antecipada do processo de *Rate-Distortion Optimization* (RDO) de *Coding Tree Units* (CTUs) e *Prediction Units* (PUs). Inicialmente, uma solução de redução do custo computacional do *High Efficiency Video Coding* (HEVC), originalmente proposta para vídeos 2D convencionais, é avaliada no contexto do V-PCC. Em seguida, uma solução especializada para o V-PCC é proposta, utilizando modelos de aprendizado de máquina treinados com dados extraídos do processo de codificação do V-PCC. Os resultados mostram que a solução proposta reduz em cerca de 60% o tempo gasto na codificação dos subfluxos do V-PCC, com um impacto médio de BD-Rate de 1,75% para atributos e 1,16% para geometria, em comparação com o *software* de referência do V-PCC, ao utilizar o modo *Random Access* (RA). Para o modo *All Intra* (AI), a redução de tempo é de cerca de 37%, com um impacto médio de BD-Rate de 0,4% para atributos e -0,13% para geometria. Esses resultados demonstram que a solução proposta obteve uma boa redução do custo computacional do V-PCC, com um impacto mínimo na eficiência de compressão, o que, em conjunto com outras técnicas, pode ser usado para viabilizar o uso em aplicações em tempo real e em dispositivos com limitações de consumo energético.

Palavras-chave: compressão de nuvens de pontos; video-based point cloud compression; aprendizado de máquina; redução de custo computacional.

ABSTRACT

REHBEIN, Gustavo Hübner. **Computational Cost Reduction of Dynamic Point Cloud Coding**. Advisor: Marcelo Schiavon Porto. 2025. 90 f. Dissertation (Masters in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2025.

Point clouds are a three-dimensional digital media format widely used in various applications, such as mapping historical objects, tele-immersive 3D videos, visualization in virtual or augmented reality, and autonomous vehicle navigation. Dynamic point clouds, which represent a sequence of point clouds, are analogous to digital 2D videos and present a large volume of data. Video-based Point Cloud Compression (V-PCC) is a dynamic point cloud compression standard developed by MPEG, which uses conventional 2D video encoders to compress the data. However, the video encoding stage of V-PCC reference software accounts for about 90% of the total computational cost of the encoding process, making V-PCC challenging for real-time applications and devices with energy constraints. This dissertation presents a solution for reducing the computational cost of V-PCC, focusing on the video encoding stage of the V-PCC substreams. Machine learning models based on decision trees are used to perform early termination of the Rate-Distortion Optimization (RDO) process for Coding Tree Units (CTUs) and Prediction Units (PUs). Initially, a computational cost reduction solution for High Efficiency Video Coding (HEVC), originally proposed for conventional 2D videos, is evaluated in the context of V-PCC. Next, a specialized solution for V-PCC is proposed, using machine learning models trained with data extracted from the V-PCC encoding process. The results show that the proposed solution reduces the time spent encoding V-PCC substreams by about 60%, with an average BD-Rate impact of 1.75% for attributes and 1.16% for geometry, compared to the V-PCC reference software, when using the Random Access (RA) mode. For the All Intra (AI) mode, the time reduction is about 37%, with an average BD-Rate impact of 0.4% for attributes and -0.13% for geometry. These results demonstrate that the proposed solution achieves a significant reduction in V-PCC computational cost with minimal impact on compression efficiency, which, together with other techniques, can be used to enable use in real-time applications and in devices with energy consumption limitations.

Keywords: point cloud compression; video-based point cloud compression; machine learning; computational cost reduction.

LISTA DE FIGURAS

Figura 1	Nuvem de pontos com zoom em detalhe para visualização dos pontos.	19
Figura 2	Exemplo de uma nuvem de pontos estática.	20
Figura 3	Exemplo de uma nuvem de pontos dinâmica.	20
Figura 4	Exemplo de uma nuvem de pontos dinamicamente adquirida.	20
Figura 5	Diagrama de funcionamento do HEVC.	24
Figura 6	Exemplo de uma CTU 64×64 e sua estrutura de árvore de codificação, dividida em CUs menores.	25
Figura 7	Tipos de particionamento de PUs no HEVC.	26
Figura 8	Nuvem de pontos projetada no plano da caixa delimitadora.	27
Figura 9	Exemplo de uma imagem de ocupação (a), imagem de geometria (b) e imagem de atributos (c), extraídos do V-PCC.	29
Figura 10	Diagrama de blocos do codificador V-PCC.	29
Figura 11	Diagrama de blocos do decodificador do V-PCC.	30
Figura 12	Métrica ponto-a-plano.	31
Figura 13	Curva de Taxa-Distorção.	32
Figura 14	Primeiro quadro de cada sequência de teste da CTC do V-PCC.	33
Figura 15	Distribuição do tempo de codificação do V-PCC para a configuração temporal (RA) em porcentagem do tempo total de codificação.	34
Figura 16	Matriz de confusão.	37
Figura 17	Exemplo de árvore de decisão.	40
Figura 18	Fluxograma da solução combinada.	63
Figura 19	Estrutura de quadros do V-PCC no modo AI.	75

LISTA DE TABELAS

Tabela 1	Sequências de teste do V-PCC.	33
Tabela 2	Porcentagem do tempo total do HEVC gasto na codificação de cada subfluxo.	35
Tabela 3	Resultados médios de redução de tempo de codificação dos subfluxos de geometria e atributo para a esquema adaptado de vídeos 2D.	52
Tabela 4	Resultados de BD-Rate de Atributo para o esquema adaptado.	53
Tabela 5	Resultados de BD-Rate de Geometria para o esquema adaptado.	54
Tabela 6	Razão de BD-Rate por redução de tempo de codificação para o esquema adaptado.	55
Tabela 7	Espaço de busca de hiperparâmetros para o ajuste dos modelos de árvores de decisão.	58
Tabela 8	Resultados da busca aleatória para cada <i>dataset</i>	59
Tabela 9	Espaço de busca de hiperparâmetros para o ajuste dos modelos de árvores de decisão.	61
Tabela 10	Resultados da busca aleatória para cada <i>dataset</i>	62
Tabela 11	Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de CTUs.	64
Tabela 12	Resultados de BD-Rate de Atributos para a abordagem especializada de interrupção antecipada de CTUs.	65
Tabela 13	Resultados de BD-Rate de Geometria para a abordagem especializada de interrupção antecipada de CTUs.	66
Tabela 14	Razão de BD-Rate e redução de tempo de codificação para a abordagem especializada de interrupção antecipada de CTUs.	67
Tabela 15	Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de PUs.	68
Tabela 16	Resultados de Attr.BD-AttrRate e Attr.BD-TotalRate para a abordagem especializada de interrupção antecipada de PUs.	69
Tabela 17	Resultados de Geom.BD-GeomRate e Geom.BD-TotalRate para a abordagem especializada de interrupção antecipada de PUs.	69
Tabela 18	Razão de BD-Rate e redução de tempo de codificação para a abordagem especializada de interrupção antecipada de PUs.	70

Tabela 19	Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de CTUs e PUs.	71
Tabela 20	Resultados de Attr.BD-AttrRate e Attr.BD-TotalRate para a solução combinada de interrupção antecipada de CTUs e PUs.	72
Tabela 21	Resultados de Geom.BD-GeomRate e Geom.BD-TotalRate para a solução combinada de interrupção antecipada de CTUs e PUs.	72
Tabela 22	Razão de BD-Rate e redução de tempo de codificação para a solução combinada de interrupção antecipada de CTUs e PUs.	73
Tabela 23	Comparação das abordagens propostas quanto à redução de tempo, impacto em BD-Rate e relação BD-Rate/Red. Tempo.	74
Tabela 24	Resultados de tempo de codificação da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.	76
Tabela 25	Resultados de BD-Rate para Atributos da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.	76
Tabela 26	Resultados de BD-Rate para Geometria da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.	77
Tabela 27	Razão de BD-Rate e redução de tempo de codificação para a solução combinada de interrupção antecipada de CTUs e PUs no modo AI do V-PCC.	77
Tabela 28	Comparação das soluções propostas com trabalhos relacionados quanto à redução de tempo, impacto em BD-Rate e relação BD-Rate por redução de Tempo.	79

LISTA DE ABREVIATURAS E SIGLAS

ADASYN	Adaptive Synthetic Sampling Method
AI	All Intra
Attr	Attribute (Atributo)
Attr.BD-AttrRate	BD-Rate considerando apenas o subfluxo de atributos
Attr.BD-TotalRate	BD-Rate considerando o bitrate total do bitstream final
AVC	Advanced Video Coding
BD-Rate	Bjontegaard Delta Rate
BOF	Blocky Occupancy Flag
Cb	Crominância azul
CNN	Convolutional Neural Network
CPU	Central Processing Unit
Cr	Crominância vermelha
CTC	Common Test Conditions
CTU	Coding Tree Unit
CU	Coding Unit
D1	Distância ponto-a-ponto
D2	Distância ponto-a-plano
FPS	Frames Per Second (Quadros por Segundo)
G-PCC	Geometry-based Point Cloud Compression
Geom	Geometry (Geometria)
Geom.BD-GeomRate	BD-Rate considerando apenas o subfluxo de geometria
Geom.BD-TotalRate	BD-Rate considerando o bitrate total do bitstream final
GPU	Graphics Processing Unit
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
ISP	Intra Sub-Partition

LiDAR	Light Detection and Ranging
Luma	Canal de luminância (Y)
ME	Motion Estimation
MIP	Matrix Intra Prediction
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MSM	Merge Skip Mode
POC	Picture Order Count
PPN	Portable Perceptron Network
PSNR	Peak Signal-to-Noise Ratio
PU	Prediction Unit
QTMT	Quadtree with nested Multi-type Tree
RA	Random Access
RAM	Random Access Memory
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RGB	Red, Green, Blue
SMOTE	Synthetic Minority Oversampling Technique
TMC2	Test Model Category 2
TotalRate	Taxa total do bitstream final
V-PCC	Video-based Point Cloud Compression
VTM	VVC Test Model
VVC	Versatile Video Coding
YCbCr	Luminância, Crominância azul, Crominância vermelha

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	16
1.1.1	Objetivo Específicos	16
1.2	Organização do Trabalho	17
2	REFERENCIAL TEÓRICO	18
2.1	Nuvens de Pontos	18
2.2	Compressão de Nuvens de Pontos	21
2.2.1	HEVC	23
2.2.2	V-PCC	26
2.3	Avaliação de Nuvens de Pontos Dinâmicas	31
2.3.1	Condições Comuns de Teste do V-PCC	32
2.4	Análise de Complexidade do V-PCC	33
2.5	Fundamentos de Aprendizado de Máquina	35
3	TRABALHOS RELACIONADOS	42
3.1	Redução de Custo Computacional em Codificadores de Vídeo	42
3.2	Redução de Custo Computacional no V-PCC	44
4	ESQUEMA PARA REDUÇÃO DO CUSTO COMPUTACIONAL DE CODIFICAÇÃO DE VÍDEOS 2D CONVENCIONAIS NO V-PCC	48
4.1	Solução para Redução do Custo Computacional do HEVC	48
4.2	Metodologia	50
4.3	Resultados Experimentais	50
5	SOLUÇÃO PARA REDUÇÃO DO CUSTO COMPUTACIONAL DO V-PCC	56
5.1	Metodologia	57
5.1.1	Abordagem de Interrupção Antecipada de CTUs	57
5.1.2	Abordagem de Interrupção Antecipada de PUs	59
5.1.3	Solução Combinada de Interrupção Antecipada de CTUs e PUs	62
5.2	Resultados para Interrupção Antecipada de CTUs	63
5.3	Resultados para Interrupção Antecipada de PUs	67
5.4	Resultados para Interrupção Antecipada de CTUs e PUs	70
5.4.1	Resultados para o Modo AI	74
5.4.2	Comparações com Trabalhos Relacionados	78
6	CONCLUSÃO	80
	REFERÊNCIAS	82

APÊNDICE A	TRABALHOS SUBMETIDOS E PUBLICADOS	88
A.1	Coding Efficiency and Complexity Analysis of the Geometry-based Point Cloud Encoder	88
A.2	Adaptive Complexity Control for AV1 Video Encoder Using Machine Learning	89
A.3	A Machine-Learning-Driven Fast Video-based Point Cloud Compression (V-PCC)	89
A.4	Efficiency and Complexity Analysis of Video-based and Geometry-based Point Cloud Encoders	90

1 INTRODUÇÃO

A crescente popularidade de tecnologias de captura e visualização tridimensional, como *LiDARs* e câmeras 3D, tem impulsionado o uso de nuvens de pontos em diversas aplicações, incluindo mapeamento de objetos históricos, vídeos 3D tele-imersivos, visualização em realidade virtual ou aumentada e navegação autônoma de veículos (Graziosi et al., 2020). No entanto, a transmissão e o armazenamento dessas nuvens de pontos apresentam desafios significativos devido à sua alta complexidade computacional e ao grande volume de dados gerados. Uma nuvem de pontos dinâmica, por exemplo, contendo cerca de um milhão de pontos por quadro e uma taxa de 30 quadros por segundo, exigiria uma conexão de aproximadamente 3,6 Gbps para transmissão em tempo real (Graziosi et al., 2020). Esse valor é significativamente superior à maior velocidade média de conexão obtida por país em 2024 (Taylor, 2024), o que torna inviável a transmissão direta desse tipo de mídia sem compressão. Nesse contexto, a compressão de nuvens de pontos torna-se essencial para viabilizar o uso eficiente desse tipo de mídia.

O grupo *Moving Picture Experts Group* (MPEG) lançou, em 2017, uma chamada de propostas para a criação de padrões de codificação de nuvens de pontos (Schaefer, 2017). A partir das propostas enviadas, duas soluções foram selecionadas para iniciar o processo de padronização: o *Video-based Point Cloud Compression* (V-PCC), voltado para nuvens de pontos dinâmicas, e o *Geometry-based Point Cloud Compression* (G-PCC), destinado a nuvens de pontos estáticas e dinamicamente adquiridas. O V-PCC faz uma etapa de planificação das nuvens de pontos, projetando-as para o espaço 2D através de três subfluxos de vídeo: o Mapa de Atributos, que armazena informações sobre a aparência dos pontos (tipicamente, dados de cor RGB); o Mapa de Geometria, que guarda informações sobre a profundidade dos pontos; e o Mapa de Ocupação, que contém informação binária indicando onde há pontos válidos no quadro 2D. Esses subfluxos são codificados utilizando codificadores de vídeo 2D convencionais, como o *High Efficiency Video Coding* (HEVC) (Sullivan et al., 2012a). Essa abordagem permite a utilização de tecnologias já existentes para a compressão de conteúdo 2D, facilitando a adoção do padrão em sistemas existentes. A implemen-

tação de referência do padrão V-PCC é chamada de *Test Model Category 2* (TMC2) (MPEG, 2024).

No entanto, a codificação dos três subfluxos de vídeo do V-PCC representa uma parte significativa do custo computacional total do processo de codificação, com cerca de 90% do tempo total de codificação sendo gasto nessa etapa (Rehbein et al., 2024) (Santos et al., 2025). Esse alto custo computacional torna o V-PCC desafiador para aplicações em tempo real e em dispositivos com restrições de energia, como dispositivos móveis e sistemas embarcados. Diante disso, a redução do custo computacional do V-PCC é um tema de pesquisa relevante e necessário para viabilizar o uso eficiente desse padrão em cenários práticos.

O uso de modelos de aprendizado de máquina pode oferecer uma abordagem promissora para a redução do custo computacional do V-PCC. Esses modelos podem ser treinados para tomar decisões informadas durante o processo de codificação, otimizando a escolha de modos de predição, particionamento e outros parâmetros do codificador. A utilização de aprendizado de máquina pode permitir uma adaptação dinâmica às características específicas das nuvens de pontos, reduzindo o custo computacional sem comprometer significativamente a eficiência da compressão.

Diante desse contexto, esta dissertação propõe o desenvolvimento de uma solução para a redução do custo computacional do *software* de referência do V-PCC, com foco na etapa de codificação de vídeo dos subfluxos do V-PCC. A proposta é utilizar modelos de aprendizado de máquina baseados em árvores de decisão para a tomada de decisão durante o processo de codificação das nuvens de pontos dinâmicas. A ideia é explorar o uso de modelos de aprendizado de máquina aplicados no processo de codificação, visando reduzir o custo computacional e viabilizar o uso eficiente do V-PCC em aplicações práticas.

1.1 Objetivos

O objetivo geral desta dissertação é reduzir o custo computacional do *software* de referência do V-PCC, com foco na etapa de codificação de vídeo dos subfluxos do V-PCC, utilizando modelos de aprendizado de máquina baseados em árvores de decisão para a tomada de decisão durante o processo de codificação das nuvens de pontos dinâmicas.

1.1.1 Objetivo Específicos

- Testar a hipótese de que soluções existentes na literatura para redução do custo computacional do *software* de referência do HEVC, elaboradas para o contexto de vídeos 2D convencionais, podem ser aplicadas no contexto da compressão de nuvens de pontos no V-PCC.

- Criar *datasets* com dados extraídos do processo de codificação de nuvens de pontos dinâmicas no *software* de referência do V-PCC para treinamento de modelos de aprendizado de máquina.
- Desenvolver soluções de redução do custo computacional para o *software* de referência do V-PCC, especializadas para o contexto de nuvens de pontos, a partir do uso de modelos de aprendizado de máquina para tomada de decisão durante o processo de codificação dos subfluxos do V-PCC.

1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma: No Capítulo 2, são apresentados os conceitos fundamentais sobre nuvens de pontos, compressão de nuvens de pontos, fundamentos de aprendizado de máquina e detalhes dos padrões HEVC e V-PCC. No Capítulo 3, são discutidos os principais trabalhos relacionados à redução de custo computacional em codificadores de vídeo e no V-PCC. No Capítulo 4, é apresentada a análise e adaptação de uma solução de redução de custo computacional originalmente proposta para vídeos 2D convencionais, avaliando sua aplicação no contexto do V-PCC. No Capítulo 5, são detalhadas as soluções especializadas propostas para redução do custo computacional do V-PCC. No Capítulo 6, são apresentadas as conclusões do trabalho.

2 REFERENCIAL TEÓRICO

Nesse capítulo será apresentado um breve referencial teórico sobre nuvens de pontos, compressão de nuvens de pontos, fundamentos de aprendizado de máquina e detalhes dos padrões HEVC e V-PCC. Esses conceitos são fundamentais para a compreensão do trabalho apresentado nesta dissertação, pois fornecem a base teórica necessária para entender os desafios e as soluções propostas para a redução do custo computacional do V-PCC.

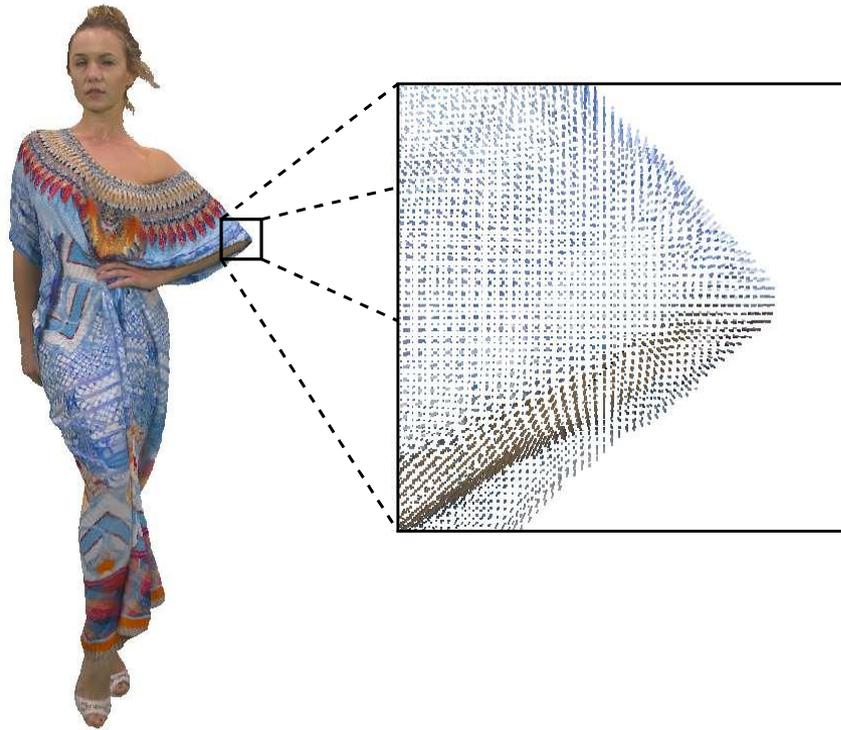
2.1 Nuvens de Pontos

As nuvens de pontos são um formato de representação de conteúdo tridimensional em formato digital. Esse formato de mídia é utilizado em diversos cenários, tais como mapeamento de objetos históricos, vídeos 3D tele-imersivos, visualização de conteúdo em realidade virtual ou aumentada e navegação autônoma de veículos (Schwarz et al., 2018). Além disso, encontram aplicação em várias áreas da indústria, como digitalização e modelagem 3D, monitoramento ambiental, agricultura, silvicultura, entre outras atividades (Cao, 2021).

Uma nuvem de pontos é constituída por um conjunto de pontos em um espaço 3D. Cada ponto está associado a uma coordenada cartesiana tridimensional, podendo também estar vinculado a informações adicionais, como cor, refletância, normal ou outras características (Cao, 2021). A Figura 1 mostra um exemplo de uma nuvem de pontos, com zoom em uma área da superfície para visualização dos pontos.

A captura de nuvens de pontos é realizada principalmente por dois tipos de dispositivos: *Laser scanners* ou *LiDARs* (Collis, 1970), que capturam a distância entre o sensor e a superfície do objeto, e câmeras ou sensores do tipo RGB-D, também chamados de câmeras 3D ou *Depth Cameras*. Nesses dispositivos, além da informação de cor RGB, também é armazenada a informação de profundidade de cada amostra capturada (Schwarz et al., 2018). Outra possibilidade de aquisição de uma nuvem de pontos é por meio da utilização de múltiplas câmeras ou múltiplas capturas e do uso de técnicas como fotogrametria e disparidade estéreo (Schwarz et al., 2018).

Figura 1 – Nuvem de pontos com zoom em detalhe para visualização dos pontos.



Fonte: Autor.

Conforme as definições do grupo *Moving Picture Experts Group* (MPEG), as nuvens de pontos podem ser classificadas em três principais categorias: nuvens de pontos estáticas (Categoria 1), dinâmicas (Categoria 2) e dinamicamente adquiridas (Categoria 3) (Li; Gao; Gao, 2024).

Uma nuvem de pontos estática, ou da categoria 1, representa a captura de uma cena ou objeto em um único instante de tempo, de forma análoga a uma imagem 2D comum. A Figura 2 apresenta um exemplo de uma nuvem de pontos dessa categoria. Nuvens de pontos dinâmicas, ou da categoria 2, representam uma sequência de nuvens de pontos, ou a captura de uma cena ou objeto ao longo de um período de tempo, sendo análogas a um vídeo 2D digital. Por se tratar de uma sequência de nuvens de pontos, nuvens dessa categoria representam uma enorme quantidade de dados. A Figura 3 mostra um exemplo de uma nuvem de pontos dinâmica, demonstrando a mudança da nuvem ao longo da sequência. Já as nuvens dinamicamente adquiridas, ou da categoria 3, são semelhantes às nuvens estáticas; no entanto, geralmente são nuvens de pontos de larga escala, constituídas de várias capturas em diferentes pontos de um ambiente ou local. Esse tipo de nuvem de pontos é usado principalmente para o mapeamento de ambientes externos, onde o sensor é movimentado ao longo do ambiente a ser capturado, sendo a nuvem de pontos final a união dessas diversas capturas. A Figura 4 mostra um exemplo de uma nuvem de pontos dessa categoria.

Figura 2 – Exemplo de uma nuvem de pontos estática.



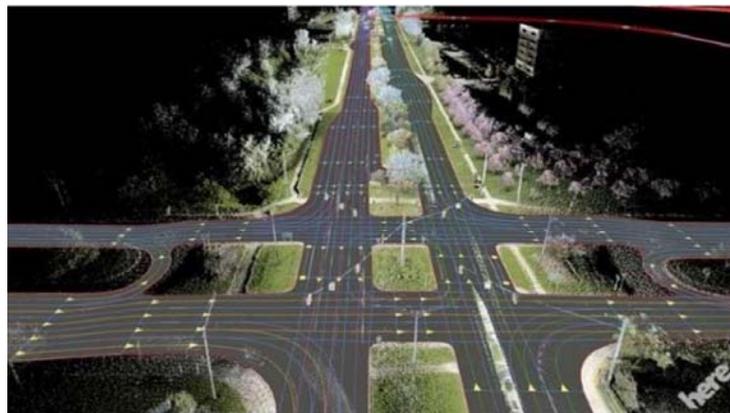
Fonte: Autor.

Figura 3 – Exemplo de uma nuvem de pontos dinâmica.



Fonte: Autor.

Figura 4 – Exemplo de uma nuvem de pontos dinamicamente adquirida.



Fonte: Autor.

2.2 Compressão de Nuvens de Pontos

A compressão de nuvens de pontos, assim como ocorre em outros tipos de mídias, como imagens e vídeos, busca reduzir a quantidade de dados necessária para representar um conteúdo. Para isso, explora-se a redundância presente nos dados, visando diminuir a quantidade de *bits* necessários para representar o conteúdo a ser comprimido. Dessa forma, é possível transmitir ou armazenar o conteúdo utilizando uma menor taxa de *bits* e menos recursos computacionais. São considerados dados redundantes aqueles que trazem informações irrelevantes ou repetidas (Gonzalez; Woods, 2010).

No contexto da compressão digital de imagens e vídeos 2D, a redundância estatística dos dados ocorre devido à correlação entre píxeis ou amostras vizinhas, que na maioria das vezes são similares ou dependentes entre si. No caso dos vídeos, também existe redundância entre quadros temporalmente vizinhos. Os símbolos usados para representar o conteúdo comprimido também podem apresentar redundância, uma vez que certos símbolos podem ser utilizados com maior frequência. Assim, existem três principais tipos de redundância em conteúdos visuais:

- **Redundância Espacial:** Refere-se à redundância estatística presente nas amostras espacialmente vizinhas, ou seja, à redundância entre as amostras presentes no mesmo quadro ou imagem. Esse tipo de redundância também é chamado de redundância intraquadro. O valor de uma amostra pode ser estimado de acordo com os valores de seus vizinhos. Assim, não é necessário armazenar o valor de cada pixel independentemente, uma vez que seu valor pode ser predito a partir dos seus vizinhos (Shi; Sun, 2000).
- **Redundância Temporal:** Refere-se à correlação estatística presente entre amostras de quadros temporalmente vizinhos de um vídeo ou sequência de imagens, sendo também chamada de redundância interquadros. Se considerarmos um vídeo em que a câmera permanece fixa ou se movimenta lentamente para outro ângulo, é esperado que grande parte dos dados permaneça similar ou apenas deslocada em relação ao quadro anterior. Assim, é possível fazer uma predição do quadro atual, ou de parte dele, utilizando amostras de outros quadros temporalmente vizinhos (Shi; Sun, 2000).
- **Redundância Entrópica:** A redundância entrópica não está diretamente relacionada ao conteúdo visual ou à informação presente nele, mas sim à representação da informação, ou seja, aos símbolos usados para representar cada informação. Considerando o número de ocorrências de cada símbolo codificado, é possível calcular a probabilidade de cada um. Se adotarmos uma representação mais curta para os símbolos com maior probabilidade de ocorrência, estaremos

aumentando a quantidade média de informação transmitida por símbolo codificado, uma vez que os símbolos mais utilizados possuem uma representação mais curta. Assim, é possível utilizar uma quantidade menor de *bits* do que em uma representação com símbolos de comprimento fixo (Shi; Sun, 2000).

Esses mesmos conceitos se aplicam a diversos tipos de mídia, como imagens, vídeos, *light fields* e também a nuvens de pontos. Nesse contexto, a redundância espacial pode ser observada na distribuição dos pontos no espaço tridimensional, onde pontos próximos tendem a ter atributos semelhantes. A redundância temporal está presente em sequências de nuvens de pontos dinâmicas (categoria 2), nas quais a geometria e os atributos de quadros consecutivos geralmente variam pouco, permitindo a predição entre quadros temporalmente vizinhos. A redundância entrópica está presente na representação dos símbolos codificados.

Nuvens de pontos representam uma enorme quantidade de dados, o que torna a transmissão e o armazenamento desse tipo de mídia uma tarefa desafiadora. Uma nuvem de pontos dinâmica, contendo aproximadamente um milhão de pontos por quadro, com uma amostragem de 30 quadros por segundo, necessitaria de uma conexão de 3,6 Gbps (Graziosi et al., 2020). Esse valor é cerca de 12 vezes superior à maior velocidade média obtida por país em 2024, conforme apresentado em Taylor (2024). Assim, o uso de compressão é essencial para viabilizar o armazenamento e a transmissão em tempo real de conteúdo baseado em nuvens de pontos.

Diante disso, o MPEG lançou, em 2017, uma chamada de propostas (Schaefer, 2017) com o objetivo de criar padrões de codificação para nuvens de pontos. Com base nas propostas enviadas, duas soluções foram selecionadas para iniciar o processo de padronização da compressão de nuvens de pontos: o *Video-based Point Cloud Compression* (V-PCC), para nuvens de pontos da categoria 2, e o *Geometry-based Point Cloud Compression* (G-PCC), para nuvens de pontos das categorias 1 e 3.

Enquanto o G-PCC trabalha diretamente sobre o conteúdo 3D para realizar a compressão, o V-PCC transforma as nuvens de pontos em uma representação 2D antes de comprimi-las. A ideia por trás dessa abordagem é que, ao converter as nuvens para o espaço bidimensional, torna-se possível utilizar tecnologias já existentes para a compressão de conteúdo 2D. No caso do V-PCC, isso permite o uso de codificadores de vídeo 2D convencionais para comprimir as nuvens de pontos convertidas.

Para planificar as nuvens de pontos, primeiramente elas são divididas ou cortadas em vários pedaços, chamados *patches*. Esses *patches* são então projetados ortogonalmente para o espaço 2D e organizados em um quadro bidimensional. No entanto, a informação de profundidade do eixo z é perdida durante essa projeção. Para permitir a reconstrução da nuvem de pontos, essa informação deve ser armazenada de alguma

forma. Para lidar com essa questão, o V-PCC utiliza vários fluxos de vídeo para representar a nuvem de pontos: o Mapa de Atributos, que armazena informações sobre a aparência dos pontos (tipicamente, dados de cor RGB); o Mapa de Geometria, que guarda informações sobre a profundidade dos pontos; e o Mapa de Ocupação, que contém informação binária indicando onde há pontos válidos no quadro 2D. Adicionalmente, informações sobre cada *patch* também são incluídas no *bitstream* final do V-PCC para permitir a reconstrução da nuvem de pontos (Graziosi et al., 2020).

O uso de três fluxos de vídeo gera algumas complicações relacionadas ao custo computacional, visto que é necessário codificar todos esses fluxos. Análises realizadas com o *software* de referência do V-PCC, o (TMC2) (MPEG, 2024), indicam que, ao utilizar o *software* de referência do *High Efficiency Video Coding* (HEVC) (Sullivan et al., 2012a) como codificador de vídeo, cerca de 90% do tempo total de codificação é gasto na codificação dos fluxos de vídeo (Rehbein et al., 2024) (Santos et al., 2025).

Assim, a elevada complexidade computacional do V-PCC representa um desafio para sua implementação em aplicações em tempo real e em dispositivos com restrições de energia. A necessidade de codificar múltiplos fluxos de vídeo impõe um alto custo de processamento, tornando essencial o desenvolvimento de estratégias para a redução do custo computacional. Dessa forma, otimizações no processo de codificação podem viabilizar o uso eficiente do V-PCC em cenários práticos, permitindo a transmissão de nuvens de pontos em tempo real e seu processamento em dispositivos móveis.

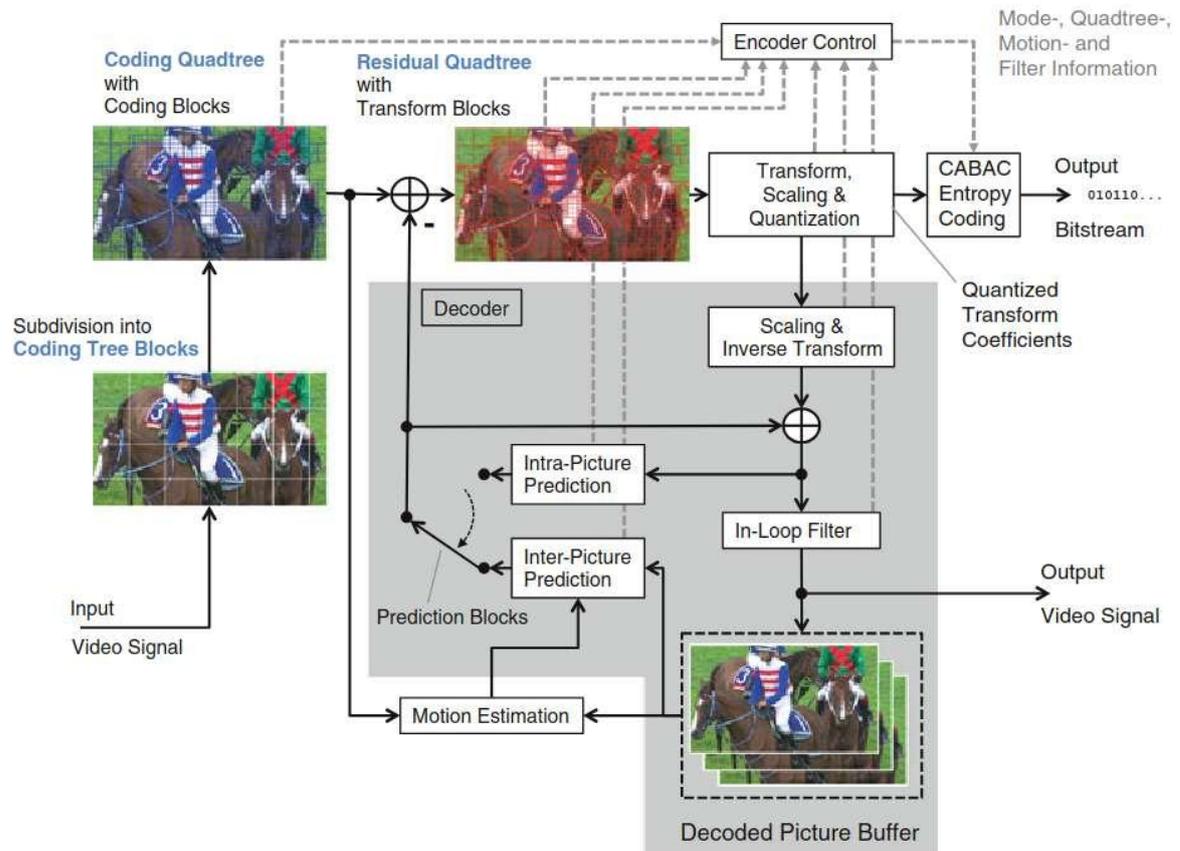
2.2.1 HEVC

High Efficiency Video Coding (HEVC), ou H.265, é um padrão de codificação de vídeo lançado em 2013 pelo *Joint Collaborative Team on Video Coding* (JCT-VC) (Sullivan et al., 2012b). O HEVC é o sucessor do padrão *Advanced Video Coding* (AVC) ou H.264 (Wiegand et al., 2003), e apresenta uma melhora de eficiência de codificação de cerca de 50% em relação ao seu antecessor (Grois; Nguyen; Marpe, 2016), ou seja, para uma mesma qualidade objetiva, o HEVC consegue uma taxa de *bits* 50% menor. Parte desses ganhos se deve ao uso de uma nova estrutura de particionamento de blocos em árvore. O *software* de referência do padrão HEVC é o *HEVC Test Model* (HM) (MPEG, 2021).

O padrão HEVC utiliza o princípio da codificação de vídeo híbrida baseada em blocos (Sze; Budagavi; Sullivan, 2014). Um diagrama que representa o funcionamento do HEVC é apresentado na Figura 5. Nesse princípio, o quadro é inicialmente dividido em blocos e, então, cada bloco é predito utilizando amostras do mesmo quadro (predição intraquadro), explorando a redundância espacial, ou amostras de outros quadros já codificados (predição interquadros), explorando a redundância temporal entre os quadros. O bloco predito é, então, comparado com o bloco original para gerar a diferença

ou resíduo de predição.

Figura 5 – Diagrama de funcionamento do HEVC.



Fonte: Sze; Budagavi; Sullivan (2014).

Após a escolha do modo de predição, o resíduo de predição é enviado para a etapa de transformada, onde é convertido para o domínio das frequências. Na etapa de quantização, os componentes de alta frequência são identificados, podendo ser atenuados ou descartados. Pois, como o olho humano é menos sensível a altas frequências, esses componentes podem ser atenuados ou removidos com pouca perda perceptível de qualidade (Ang; Ruetz; Auld, 1991).

A etapa de codificação de entropia é a última etapa da codificação de vídeo. Essa etapa é responsável por remover a redundância entrópica presente na representação dos símbolos, ou seja, por reduzir o número de *bits* necessários para representar o *bitstream* gerado pelo codificador.

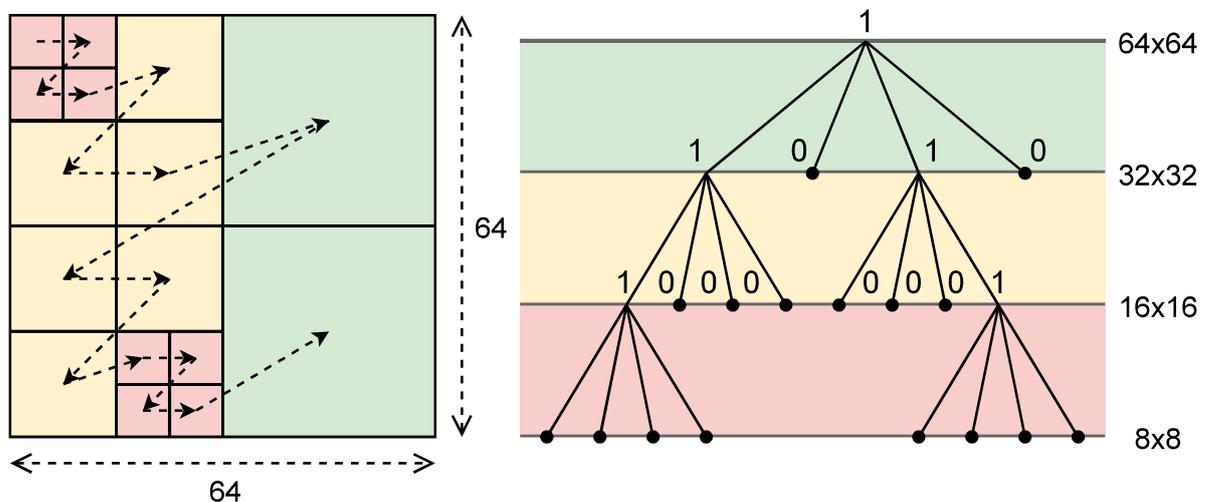
Como informações são descartadas na etapa de quantização, o resultado é um bloco diferente do original. Assim, para garantir que as amostras usadas na predição sejam idênticas às utilizadas na decodificação do vídeo codificado, são empregadas as etapas de quantização e transformada inversa para reconstruir os blocos preditos após a quantização. Esses blocos reconstruídos são os que são usados nas etapas

de predição, garantindo a consistência dos dados na codificação e decodificação dos vídeos. A etapa de filtragem *in-loop* é utilizada para remover artefatos que possam ser gerados pelo processo de codificação, como descontinuidades presentes nas bordas dos blocos.

2.2.1.1 Estrutura de Particionamento do HEVC

O HEVC particiona o quadro em diversos blocos menores, de forma recursiva, em uma estrutura de árvore. Nessa estrutura, cada unidade de particionamento básica é chamada de Unidade de Árvore de Codificação, ou *Coding Tree Unit* (CTU). Cada CTU pode ser subdividida em unidades menores chamadas Unidades de Codificação, ou *Coding Units* (CU); essa divisão segue a estrutura de particionamento de *Quadtrees*, sendo dividida em quatro partes iguais. Cada CU pode ser subdividida recursivamente até tamanhos de 8×8 para amostras de luminância e 4×4 para amostras de croma, com um tamanho máximo de 64×64 . A Figura 6 ilustra o particionamento de uma CTU 64×64 e sua estrutura de árvore de codificação correspondente. Subdivisões de CUs são indicadas por 1s, enquanto 0s denotam níveis sem subdivisões adicionais.

Figura 6 – Exemplo de uma CTU 64×64 e sua estrutura de árvore de codificação, dividida em CUs menores.

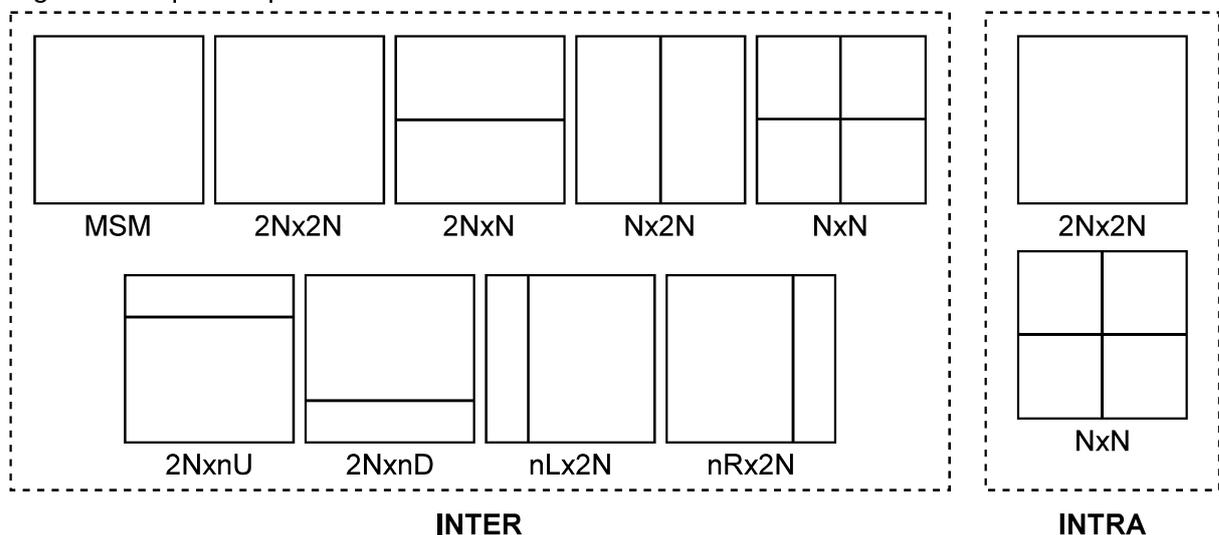


Fonte: Adaptado de Kim et al. (2012).

Cada CU pode ainda ser subdividida em duas ou mais Unidades de Predição, ou *Prediction Unit* (PU). Cada PU é predita separadamente e armazena informações sobre a predição, como vetores de movimento, modo de particionamento e modo de codificação. O HEVC suporta oito tipos de particionamento de PUs, apresentados na Figura 7, sendo dois quadráticos e seis retangulares. O particionamento $2N \times 2N$ não apresenta nenhuma subdivisão em relação à CU utilizada. O número de possibilidades de particionamentos varia de acordo com o tamanho da CU sendo testada e

o tipo de predição utilizada. Partições $N \times N$ só são testadas se a CU tem o maior tamanho de CU disponível (por padrão, 64×64), para evitar o teste de PUs com o mesmo tamanho em níveis diferentes da CTU (testar $N \times N$ seria equivalente a testar $2N \times 2N$ no nível acima da CTU) (Correa et al., 2014b). Partições que levariam a PU com dimensões menores que 4 também não são testadas ($2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$ em CUs de tamanho 8×8). Para PUs preditas com predição intraquadro, apenas os modos quadráticos estão disponíveis ($N \times N$ e $2N \times 2N$) e o modo *Merge Skip Mode* (MSM) está apenas disponível para PUs preditas com modo interquadros. O MSM permite que uma PU herde diretamente os parâmetros de movimento (vetores de movimento e referências) de blocos vizinhos já codificados, sem a necessidade de transmitir explicitamente novos vetores de movimento.

Figura 7 – Tipos de particionamento de PUs no HEVC.



Fonte: Adaptado de Correa et al. (2014b).

Todas as possibilidades de particionamento de CTU e PU são avaliadas utilizando o processo de Otimização de Taxa-Distorção (*Rate-Distortion Optimization*, RDO). O RDO é responsável por selecionar a melhor configuração de particionamento que equilibra a qualidade do vídeo reconstruído (distorção) e a taxa de *bits* necessária para representá-lo.

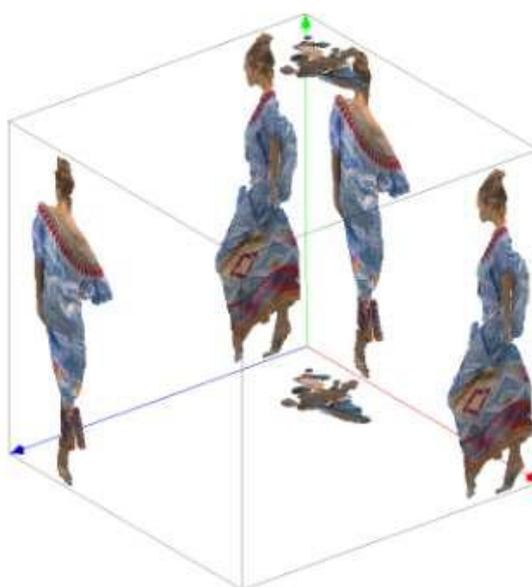
2.2.2 V-PCC

O V-PCC é um padrão de codificação de nuvens de pontos dinâmicas (categoria 2) estabelecido pelo grupo MPEG a partir de propostas enviadas em uma chamada de propostas em 2017 (Schaefer, 2017), sendo oficialmente publicado sob a norma ISO/IEC 23090-5 em 2021 (International Organization for Standardization, 2021). Den-

tre as propostas enviadas, várias utilizavam projeções de 3D para 2D. A proposta que apresentou os melhores resultados consistia em uma solução na qual a nuvem de pontos era inicialmente dividida em diversos pedaços chamados *patches* 3D (Graziosi et al., 2020), que são projetados para o espaço 2D. Usando essa abordagem, é possível utilizar técnicas existentes para compressão de imagens e vídeos 2D a fim de comprimir os *patches* da nuvem de pontos planificada, permitindo uma adoção mais fácil do padrão em sistemas existentes. A implementação de referência do padrão V-PCC (MPEG, 2024) utiliza o codificador de vídeo HEVC em sua configuração padrão, embora o V-PCC seja agnóstico quanto ao codificador utilizado (Preda, 2020).

Para criar os *patches*, inicialmente o valor da normal de cada ponto é estimado. Utilizando esse valor, cada ponto é associado a um dos seis planos de uma caixa delimitadora, definidos pela normal de cada plano, ou seja: $(1.0, 0.0, 0.0)$, $(0.0, 1.0, 0.0)$, $(0.0, 0.0, 1.0)$, $(-1.0, 0.0, 0.0)$, $(0.0, -1.0, 0.0)$ e $(0.0, 0.0, -1.0)$. O plano selecionado é aquele que possui a menor diferença em relação à normal do ponto. A Figura 8 demonstra esse processo. Após isso, os pontos são agrupados de acordo com a normal e a distância euclidiana entre eles. Os *patches* finais são criados juntando agrupamentos de pontos similares; um processo de refino também é realizado, a fim de garantir um número mínimo de *patches* para um dado número de pontos, dividindo *patches* existentes quando necessário (Preda, 2020). As projeções 2D são obtidas por meio de projeções ortogonais ao plano atribuído a cada *patch*.

Figura 8 – Nuvem de pontos projetada no plano da caixa delimitadora.



Fonte: Preda (2020)

Para tratar casos em que mais de um ponto seja projetado para a mesma coordenada 2D, o V-PCC permite o uso de duas camadas, chamadas *near layer* e *far-layer*.

Cada *patch* é projetado para ambas as *layers*, sendo que o *near layer* armazena os pontos com menor profundidade e o *far-layer* armazena os pontos sobrepostos de maior profundidade (Preda, 2020).

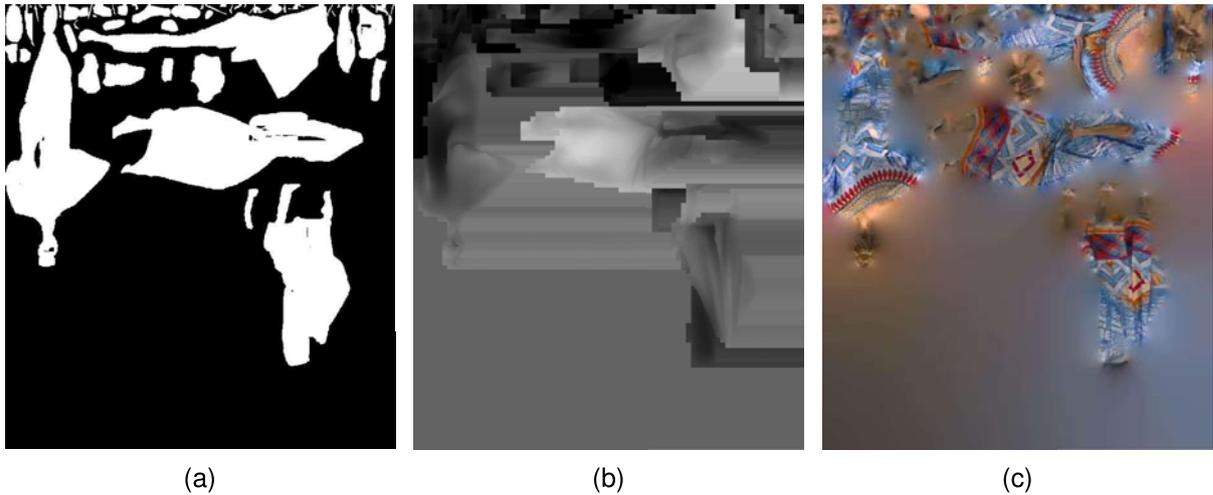
Após dividir a nuvem de pontos em diversos *patches* 3D e planificá-los para o espaço 2D, esses são organizados em uma imagem ou quadro 2D para possibilitar o uso do codificador de vídeo. Para cada *patch* é atribuída uma posição no quadro, buscando uma organização eficiente dos *patches*. Essa organização é obtida por meio de uma busca exaustiva, que posiciona os *patches* em um quadro 2D, garantindo que não haja sobreposição e respeitando uma distância mínima entre cada *patch*. Cada *patch* é inserido no quadro de maneira iterativa, de acordo com seu índice. Caso não haja espaço disponível no quadro para o próximo *patch*, a altura do quadro pode ser aumentada. A largura e a altura iniciais do quadro são definidas de acordo com o tamanho da nuvem de pontos a ser comprimida. O V-PCC também permite que cada *patch* seja rotacionado em 0, 90, 180 ou 270 graus, além da imagem espelhada de cada rotação, para permitir um posicionamento mais flexível dos *patches* (Preda, 2020).

Para permitir que os *patches* permaneçam temporalmente consistentes dentro de um grupo de quadros, o V-PCC adota uma estratégia para identificar *patches* globais e posicioná-los na mesma localização em diferentes quadros do grupo. Dessa forma, garante-se a consistência temporal para *patches* presentes em vários quadros vizinhos (Preda, 2020).

Como a informação de profundidade é perdida ao projetar os *patches* para o espaço 2D, o V-PCC utiliza dois subfluxos de vídeo para representar a nuvem planificada: um para os atributos da nuvem de pontos, chamado mapa de atributos, que armazena informações sobre a aparência ou cor de cada ponto, geralmente em RGB (*Red, Green, Blue*); e outro subfluxo para armazenar a informação de profundidade de cada ponto. Adicionalmente, o V-PCC também utiliza outro subfluxo contendo informação binária que indica se determinada posição no quadro 2D possui pontos válidos, chamado mapa de ocupação. A Figura 9 apresenta um exemplo de um quadro planificado e seus três subfluxos correspondentes.

A Figura 10 mostra as etapas do processo de codificação do padrão V-PCC. Após dividir a nuvem de pontos em *patches*, é feita a organização dos *patches* dentro do quadro 2D. Inicialmente, as imagens ou o vídeo do mapa de ocupação são gerados usando as informações e a organização dos *patches*. O mapa de ocupação, então, é codificado utilizando o codificador de vídeo. Em seguida, é feita a geração da imagem do mapa de geometria. A fim de aumentar a eficiência de codificação dos subfluxos, o fundo das imagens de geometria e de atributos é preenchido para suavizar a transição de cores entre os *patches* e o fundo da imagem, como visível nas Figuras 9b e 9c. Esse processo é realizado após a geração da imagem do mapa de geometria e

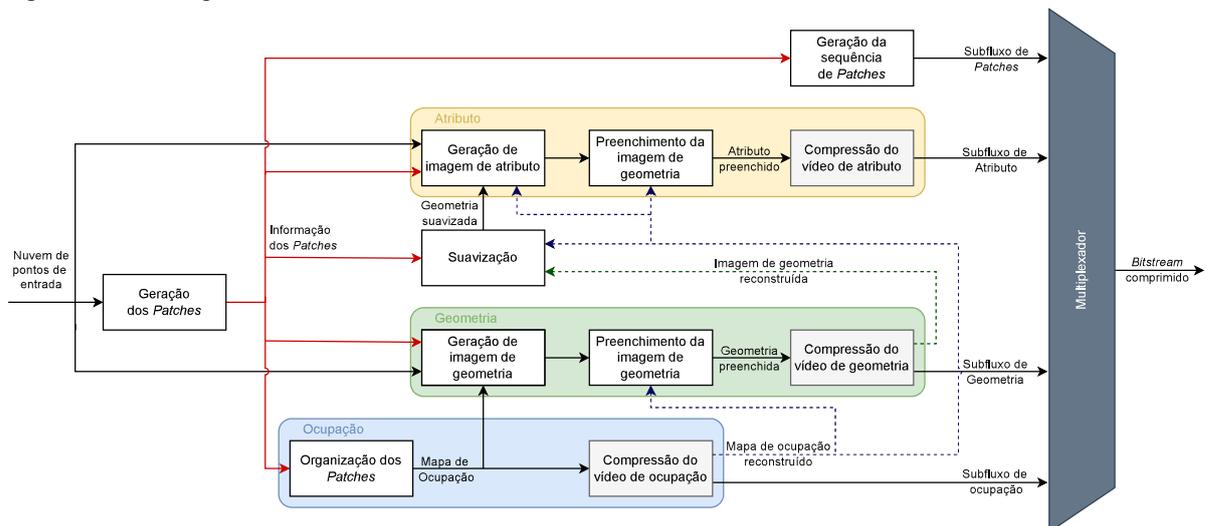
Figura 9 – Exemplo de uma imagem de ocupação (a), imagem de geometria (b) e imagem de atributos (c), extraídos do V-PCC.



Fonte: Autor.

também após a geração da imagem do mapa de atributos. Após a codificação do subfluxo do mapa de geometria, há uma etapa de suavização que busca aliviar potenciais descontinuidades que possam aparecer nas bordas dos *patches* devido a artefatos de compressão. Isso é feito movendo os pontos das bordas dos *patches* para mais próximos dos centroides dos seus pontos vizinhos mais próximos (Preda, 2020).

Figura 10 – Diagrama de blocos do codificador V-PCC.



Fonte: Adaptado de Preda (2020).

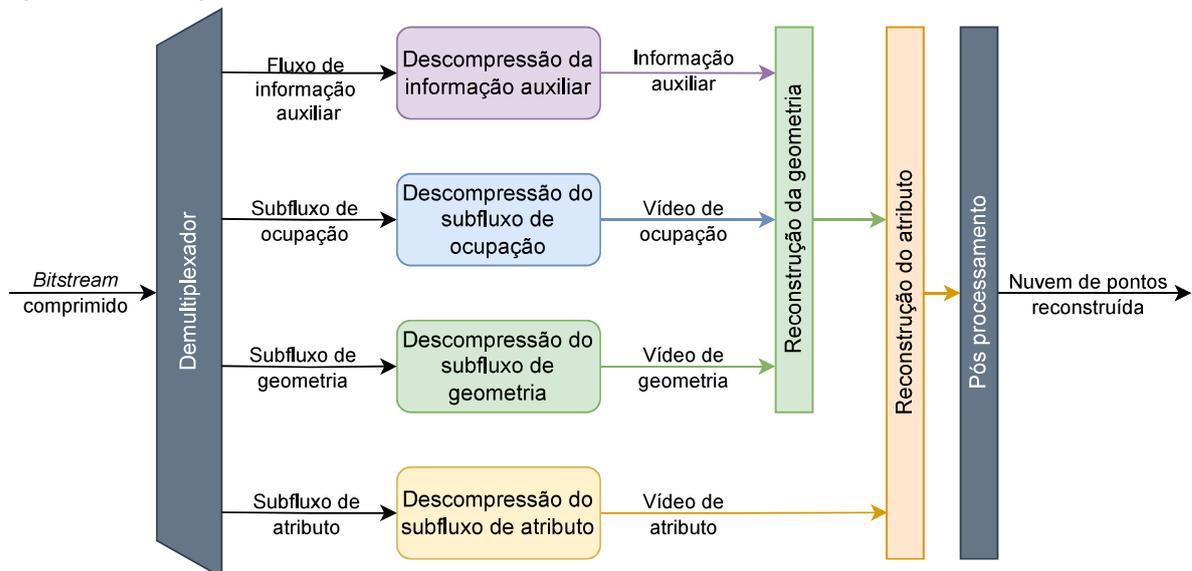
Para a geração da imagem do mapa de atributos, é utilizado o mapa de geometria reconstruído e suavizado pela etapa de suavização para calcular as novas cores que serão associadas aos pontos reamostrados ou reconstruídos. Isso é necessário, pois

o processo de codificação e suavização pode ter causado mudanças nas posições dos pontos em relação à nuvem de pontos original. Após essa etapa, o fundo da imagem do mapa de atributos é preenchido na etapa de *padding* e, em seguida, comprimido pelo codificador de vídeo.

Para permitir que a nuvem de pontos seja reconstruída, além de transmitir os subfluxos de ocupação, geometria e atributos, informações adicionais sobre os *patches* também devem ser transmitidas. Essas informações incluem: o plano de projeção, a caixa limitadora 2D da nuvem de pontos, a posição 3D dos *patches*, a segmentação dos *patches* e informações de projeção. Esses dados são comprimidos utilizando codificadores de entropia (Li; Gao; Gao, 2024).

A Figura 11 apresenta o processo de decodificação do V-PCC. Inicialmente, os três subfluxos de vídeo são extraídos do *bitstream*, juntamente com as informações auxiliares sobre os *patches*. Em seguida, cada subfluxo é descomprimido utilizando o decodificador de vídeo ou o decodificador de entropia. Para reconstruir a informação de geometria da nuvem de pontos, são utilizadas as informações sobre os *patches*, o mapa de ocupação e o mapa de geometria. Com a geometria reconstruída, o mapa de atributos é utilizado, junto com as informações sobre os *patches*, para recuperar a informação de atributo de cada ponto. Adicionalmente, uma etapa de pós-processamento é empregada para aprimorar a qualidade da nuvem de pontos reconstruída (Li; Gao; Gao, 2024).

Figura 11 – Diagrama de blocos do decodificador do V-PCC.



Fonte: Adaptado de Preda (2020).

2.3 Avaliação de Nuvens de Pontos Dinâmicas

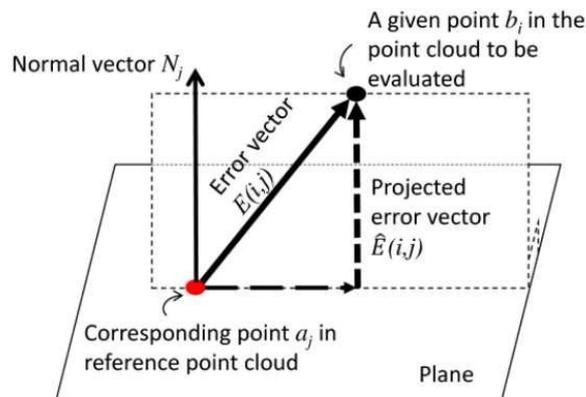
Para avaliar a qualidade objetiva de nuvens de pontos dinâmicas, são utilizadas métricas diferentes para geometria e atributo. Para geometria, são utilizadas as métricas ponto-a-ponto (D1) e ponto-a-plano (D2); para atributo, é utilizada a métrica de *Peak Signal-to-Noise Ratio* (PSNR) sobre os canais de cor YCbCr (Luminância, Crominância azul e Crominância vermelha) (MPEG, 2020).

A métrica D1 é calculada identificando os pontos correspondentes na nuvem de pontos original e na nuvem de pontos comprimida, calculando a distância ou erro entre ambos. Esse processo é repetido para todos os pontos da nuvem de pontos. D1 é dado pela média dos erros considerando todos os pontos, conforme a Equação (1), onde A é a nuvem de pontos original, B é a nuvem de pontos comprimida, NP_B é o número de pontos em B e $erro_{A,B}(i)$ é o vetor de distância do ponto i entre as nuvens A e B .

$$D1_{A,B} = \frac{1}{NP_B} \sum_{i=0}^{NP_B} erro_{A,B}(i) \quad (1)$$

A métrica D2 é semelhante à D1, porém o erro de cada ponto é obtido considerando o vetor normal do ponto da nuvem original e calculando a distância do ponto na nuvem de pontos comprimida em relação ao plano definido por esse vetor normal, conforme exemplificado na Figura 12.

Figura 12 – Métrica ponto-a-plano.



Fonte: Preda (2020).

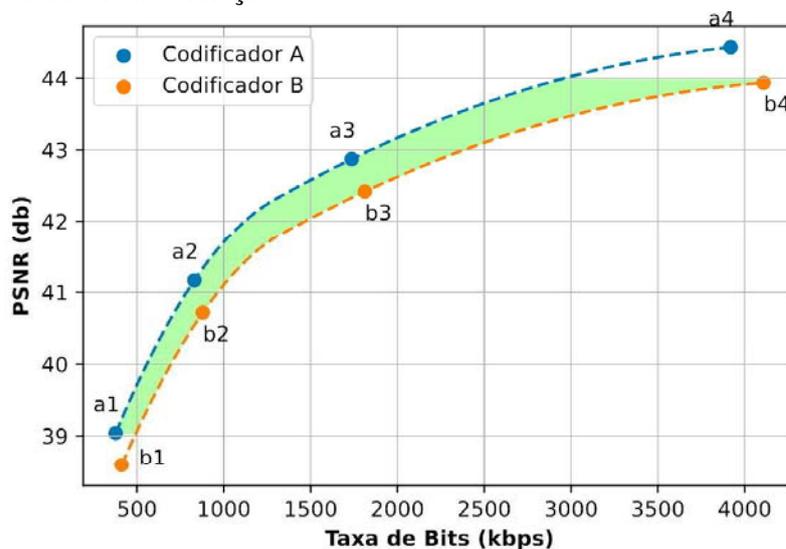
Para os atributos de cor, o erro médio quadrático (*Mean Squared Error* — MSE) de cada canal de cor é calculado no espaço de cores YCbCr. O MSE é calculado de forma simétrica, ou seja, de A em relação a B e de B em relação a A , sendo A a nuvem de pontos original e B a nuvem de pontos reconstruída. A comparação que apresenta o maior erro é a utilizada. Em seguida, o PSNR é calculado considerando

o MSE simétrico, conforme a Equação (2), onde p é o valor constante de pico definido para cada nuvem de pontos de referência.

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

Para avaliar a diferença de eficiência de codificação entre dois codificadores ou configurações de um codificador de nuvens de pontos, é utilizada a métrica de *Bjontegaard Delta Rate* (BD-Rate) (Bjontegaard, 2001). Essa métrica permite avaliar a mudança na taxa de *bits* entre dois codificadores enquanto se mantém a mesma métrica de qualidade objetiva. Para isso, é gerada uma curva a partir de quatro ou mais amostras de taxa-distorção de cada codificador ao longo de diferentes configurações de taxa de *bits*. O BD-Rate é um valor em porcentagem relacionado à área entre as duas curvas de taxa-distorção, obtido por meio da integralização dessas curvas, conforme exemplificado na Figura 13 pela área em verde entre as curvas. Valores positivos de BD-Rate indicam uma perda de eficiência de codificação, enquanto valores negativos indicam um ganho de eficiência de codificação.

Figura 13 – Curva de Taxa-Distorção.



Fonte: Autor.

2.3.1 Condições Comuns de Teste do V-PCC

O grupo MPEG estabelece regras sobre como os testes devem ser realizados e quais sequências de nuvens de pontos dinâmicas devem ser utilizadas nesses testes. Essas regras são chamadas de Condições Comuns de Teste (*Common Test Conditions* — CTC) e estão definidas em MPEG (2020).

A Tabela 1 apresenta as sequências de teste a serem usadas no V-PCC. São definidas sete sequências de teste, sendo cinco com precisão de geometria de 10 *bits*

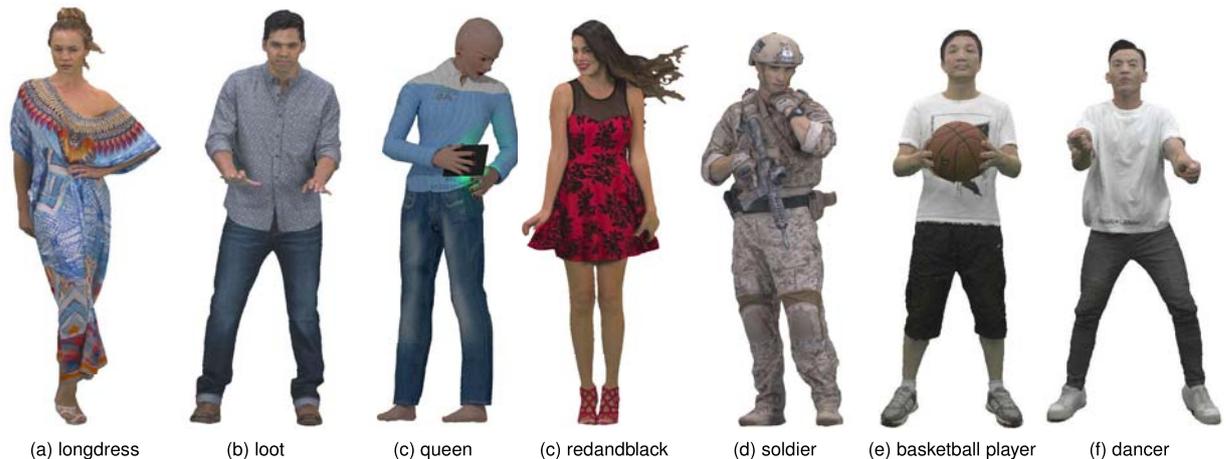
(*Queen, Loot, Red and Black, Soldier e Longdress*) e duas com precisão de geometria de 11 *bits* (*Basketball player e Dancer*). Essas sequências são classificadas nas categorias A, B e C, que descrevem o nível de complexidade das nuvens (A com a menor complexidade e C com a maior complexidade, respectivamente). O primeiro quadro de cada sequência é apresentado na Figura 14 (a) – (f).

Tabela 1 – Sequências de teste do V-PCC.

Classe	Nome da sequência	Quadros	FPS	Num. Pontos	Precisão	Valor máximo	Atributos
A	Queen	250	50	~1,000,000	10 bit	1023	R,G,B
	8i VFB – Loot	300	30	~780,000			
	8i VFB – Red and Black			~700,000			
8i VFB – Soldier	~1,500,000						
B	8i VFB – Long dress	64		~800,000	11 bit	2047	
C	basketball_player_vox11			~2,900,000			
	dancer_vox11			~2,600,000			

Fonte: MPEG (2020)

Figura 14 – Primeiro quadro de cada sequência de teste da CTC do V-PCC.



Fonte: MPEG (2020)

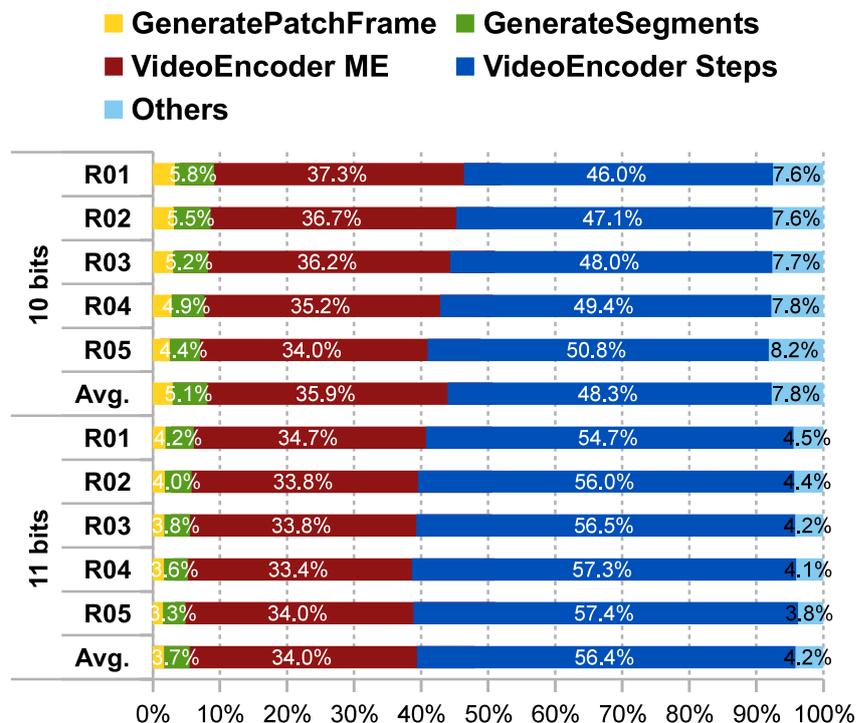
2.4 Análise de Complexidade do V-PCC

Para melhor identificar a distribuição de complexidade do V-PCC, foi realizada uma análise utilizando o *software GNU profiling* (G-Prof) na versão 2.9.1 (Fenlason, 2024). Nessa análise, foram utilizadas as sequências de teste de 10 e 11 *bits* disponibilizadas nas CTC do V-PCC (MPEG, 2020). Esses experimentos foram realizados com a versão 22.1 do TMC2, para a configuração temporal *Random Access* (RA) e para cada uma das cinco configurações de taxa de *bits* do TMC2: r1, r2, r3, r4 e r5, sendo r1 e r5 as configurações com menor e maior taxa de *bits*, respectivamente.

A Figura 15 apresenta um gráfico com os resultados médios de custo computacional para cada configuração de taxa de *bits* (exibida em porcentagem do tempo total de codificação) para nuvens de pontos dinâmicas de 10 e 11 *bits* de precisão. Estão destacadas apenas as principais etapas do processo de codificação: *GenerateSegments*, *GeneratePatchFrame* e *VideoEncoder*. O rótulo “*Others*” agrupa o tempo gasto com o *bitstream* e outras etapas de codificação.

- *GenerateSegments*: Etapa para gerar os *patches* da nuvem de pontos.
- *GeneratePatchFrame*: Etapa para organizar os *patches* de forma eficiente.
- *VideoEncoder ME*: Etapa de estimação de movimento (*Motion Estimation – ME*) da codificação, utilizando o codificador de vídeo, que por padrão é o HEVC.
- *VideoEncoder Steps*: Demais etapa da codificação de vídeo dos subfluxos.
- *Others*: Combina o tempo de codificação de outras funções, como carregamento da nuvem de pontos, geração do *bitstream* e mapa de ocupação.

Figura 15 – Distribuição do tempo de codificação do V-PCC para a configuração temporal (RA) em porcentagem do tempo total de codificação.



Fonte: Santos et al. (2025).

Observando os resultados da Figura 15, nota-se que os processos relacionados à codificação de vídeo (*VideoEncoder ME* e *VideoEncoder Steps*) correspondem à

maior fatia de custo computacional em todos os casos, sendo ainda mais expressivos para nuvens de pontos de 11 *bits*, atingindo uma média geral de cerca de 90% do tempo total de codificação do V-PCC. Enquanto isso, os processos relacionados ao TMC2 (como divisão dos *patches*, criação dos subfluxos e demais etapas) consomem apenas uma pequena parcela do tempo total de codificação. Assim, soluções para a redução de custo computacional focadas nas etapas de codificação de vídeo têm maior potencial de impacto no custo computacional do V-PCC.

Também foi realizada uma análise para identificar a distribuição do tempo de codificação entre cada subfluxo de vídeo (Ocupação, Geometria e Atributo). Foi analisado o tempo de codificação de cada subfluxo para cada sequência de teste da CTC do V-PCC. Os resultados médios para cada taxa de *bits* são apresentados na Tabela 2. Os resultados mostraram que os subfluxos de geometria e atributo correspondem, em média, a 99,18% do tempo total gasto com codificação de vídeo. Por outro lado, a codificação do mapa de ocupação corresponde a uma parcela mínima do tempo, com valor médio de 0,82% do tempo gasto pelo HEVC no V-PCC. Assim, soluções para redução de custo computacional focadas nos subfluxos de geometria e atributo têm maior potencial de impacto na diminuição do tempo total de codificação do V-PCC.

Tabela 2 – Porcentagem do tempo total do HEVC gasto na codificação de cada subfluxo.

Configuração de Taxa	Ocupação	Geometria	Atributo
r1	0,59%	54,89%	44,52%
r2	0,57%	55,10%	44,33%
r3	0,53%	54,05%	45,42%
r4	0,49%	53,36%	46,15%
r5	1,95%	50,56%	47,49%
Média	0,82%	53,59%	45,58%

2.5 Fundamentos de Aprendizado de Máquina

Aprendizado de máquina é um subcampo da ciência da computação que busca a construção de algoritmos capazes de aprender a solucionar problemas a partir de um conjunto de dados de exemplo. A partir desses dados de exemplo, um modelo estatístico é gerado utilizando algum algoritmo de aprendizado de máquina. Os dados de exemplo, ou *dataset*, são constituídos por diversos exemplos, cada um com uma quantidade determinada de características ou *features* que descrevem o problema a ser resolvido de alguma forma. O objetivo dos modelos de aprendizado de máquina é prever um valor numérico, categórico ou outro tipo de dado a partir dos dados de entrada.

Existem três principais tipos de aprendizado de máquina (Burkov, 2019), sendo eles:

- **Aprendizado Supervisionado:** Nesse tipo de aprendizado de máquina, os valores-alvo já estão presentes no *dataset* de treinamento. O objetivo dos algoritmos de aprendizado supervisionado é gerar um modelo que faça a predição do valor-alvo a partir do vetor de *features*.
- **Aprendizado Não Supervisionado:** Nesse tipo, os valores-alvo não são conhecidos na etapa de treinamento dos modelos. Assim, o objetivo dos algoritmos de aprendizado não supervisionado é gerar um modelo para categorizar ou atribuir valores aos dados de entrada de alguma forma, possibilitando sua utilização na resolução de algum problema.
- **Aprendizado por Reforço:** Nesse tipo de aprendizado, o algoritmo aprende realizando ações de acordo com os dados de entrada. Para cada ação realizada, é atribuída uma recompensa. O objetivo é maximizar as recompensas ao longo das ações executadas.

Dependendo do tipo do valor-alvo, problemas resolvidos com aprendizado de máquina podem ser classificados como de regressão ou de classificação. Em problemas de classificação, os valores-alvo pertencem a um conjunto finito de dados, ou seja, os dados são categorizados de acordo com um número limitado de categorias ou rótulos. Na regressão, o valor-alvo é um número real e o objetivo do modelo é estimar esse valor minimizando o erro. Como esta dissertação utilizará algoritmos de aprendizado de máquina de classificação, esse tipo de algoritmo será o foco deste capítulo.

Para treinar um modelo de classificação, é necessária a criação de um *dataset* com *features* relevantes para o problema que se deseja resolver. Em problemas relacionados a codificadores de vídeo ou nuvens de pontos, essas *features* são normalmente coletadas durante o processo de codificação de sequências de teste. Essas *features* incluem informações sobre o vídeo ou bloco sendo codificado, extraídas diretamente do processo de codificação ou calculadas a partir do conteúdo a ser codificado.

Para avaliar um modelo de aprendizado de máquina supervisionado de classificação, existem diversas métricas disponíveis. A matriz de confusão é uma forma de sumarizar os resultados obtidos por um modelo de classificação, apresentando os valores preditos em um eixo e os valores reais no outro, conforme ilustrado na Figura 16, onde os exemplos são classificados em *spam* ou *não-spam* e relacionados com a classe real. Nessa matriz, são apresentados quatro tipos de valores:

- **Verdadeiros positivos:** Exemplos positivos que foram classificados corretamente como positivos.
- **Falsos positivos:** Exemplos negativos que foram incorretamente classificados como positivos.

- **Verdadeiros Negativos:** Exemplos negativos que foram corretamente classificados como negativos.
- **Falsos Negativos:** Exemplos positivos que foram incorretamente classificados como negativos.

Figura 16 – Matriz de confusão.

		Valores preditos	
		Spam	Não-spam
Valores reais	Spam	Verdadeiro Positivo 60	Falso Negativo 20
	Não-spam	Falso Positivo 15	Verdadeiro Negativo 75

Fonte: Autor.

A partir desses valores, é possível calcular outras métricas relacionadas à qualidade do modelo, como *Precision* e *Recall*. O *Precision* indica a razão de verdadeiros positivos (*True Positives*, TP) sobre o número total de predições positivas (TP + *False Positives*, FP), conforme apresentado na Equação (3).

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

O *Recall* refere-se à razão de verdadeiros positivos sobre o número total de exemplos positivos no *dataset*. A Equação (4) apresenta a fórmula para o cálculo do *Recall*.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

A partir do *Precision* e do *Recall*, é possível calcular o *F1-score*, conforme apresentado na Equação (5). O *F1-score* é a média harmônica entre o *Precision* e o *Recall* e representa uma medida da acurácia do modelo.

$$f1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Utilizando essas métricas, é possível avaliar de forma abrangente o desempenho de modelos de classificação em problemas relacionados à codificação de vídeos ou nuvens de pontos. A matriz de confusão fornece uma visão detalhada sobre os acertos

e erros do modelo, enquanto métricas como *Precision*, *Recall* e *F1-score* permitem quantificar a qualidade das predições em diferentes aspectos, como a capacidade de identificar corretamente exemplos positivos e minimizar falsos positivos ou negativos. Dessa forma, essas métricas são fundamentais para ajustar e validar modelos de aprendizado de máquina, garantindo que eles atendam aos requisitos de precisão e eficiência necessários para aplicações práticas.

Os algoritmos de treinamento de modelos de aprendizado de máquina possuem diversos ajustes que alteram certas características de como o modelo é treinado. Esses ajustes são chamados de hiperparâmetros. Encontrar um ajuste de hiperparâmetros adequado pode melhorar significativamente a acurácia do modelo treinado para aquele problema e também pode ajudar a evitar problemas como *underfitting* e *overfitting*.

Existem duas principais técnicas de busca de ajustes de hiperparâmetros (Burkov, 2019): *Grid search* e *Random search*. Para ambas as técnicas, são definidos os hiperparâmetros a serem testados e o conjunto de valores para cada hiperparâmetro. No *Grid search*, a busca é realizada testando todas as combinações possíveis de ajustes de hiperparâmetros; cada ajuste é avaliado usando um *dataset* de treino e teste, e os resultados do modelo treinado com esses hiperparâmetros são avaliados por alguma métrica, como o *F1-score*. Ao final, o ajuste com o melhor resultado é selecionado para o treinamento do modelo final.

Dependendo do tamanho do *dataset* de treino, do número de hiperparâmetros e da quantidade de valores para cada um, o uso de *grid search* pode se tornar proibitivo (Burkov, 2019). Nesses casos, torna-se mais adequado o uso de *random search*. No *random search*, além dos hiperparâmetros e valores, também é indicado o número total de iterações. Em vez de testar todas as combinações possíveis, as combinações de hiperparâmetros são selecionadas aleatoriamente e o número de iterações limita o total de combinações testadas, permitindo uma busca de hiperparâmetros mais curta.

Se for usada sempre a mesma subdivisão de dados em treino e validação, há o risco de sobreajustar o modelo para aquela subdivisão específica, prejudicando a capacidade do modelo de generalizar predições. Técnicas como validação cruzada auxiliam no tratamento desse problema. A validação cruzada com *k-folds* divide o conjunto de dados de treino em k partes de tamanho igual; $(k - 1)$ partes são utilizadas para treinar o modelo e a parte restante para avaliá-lo. Esse processo é repetido k vezes, alternando a subdivisão utilizada para avaliação do modelo (Burkov, 2019).

Datasets desbalanceados, ou seja, com uma quantidade majoritariamente maior de exemplos de uma classe em relação às demais, podem criar um viés nos modelos treinados. Esse desbalanceamento faz com que o modelo favoreça a classe majoritária, prejudicando a capacidade de identificar corretamente exemplos da classe minoritária. Para tratar esse problema, existem técnicas como *Oversampling* e *Under-*

sampling, que ajustam o conjunto de treino para evitar o desbalanceamento.

No *Oversampling*, o número de exemplos da classe minoritária é aumentado por meio da duplicação de exemplos existentes ou da criação de novos exemplos sintéticos a partir das *features* dos exemplos já presentes, utilizando técnicas como *Synthetic Minority Oversampling Technique* (SMOTE) (Chawla et al., 2002) ou *Adaptive Synthetic Sampling Method* (ADASYN) (He et al., 2008). O *Undersampling* faz o oposto do *Oversampling*: nesse método, exemplos da classe majoritária são descartados aleatoriamente até que se atinja um balanceamento desejado (Burkov, 2019).

Dentre os diversos algoritmos de aprendizado de máquina supervisionado para classificação, as árvores de decisão se destacam por possuírem uma etapa de inferência simples, ou seja, os modelos treinados realizam as previsões sem demandar muitos recursos computacionais. Essa característica torna as árvores de decisão interessantes para casos em que o objetivo é reduzir o custo computacional de alguma tomada de decisão. Árvores de decisão são facilmente interpretáveis e não exigem normalização das *features*. No entanto, podem sofrer de *overfitting*, especialmente quando muito profundas, e são sensíveis a pequenas variações nos dados de entrada (Burkov, 2019).

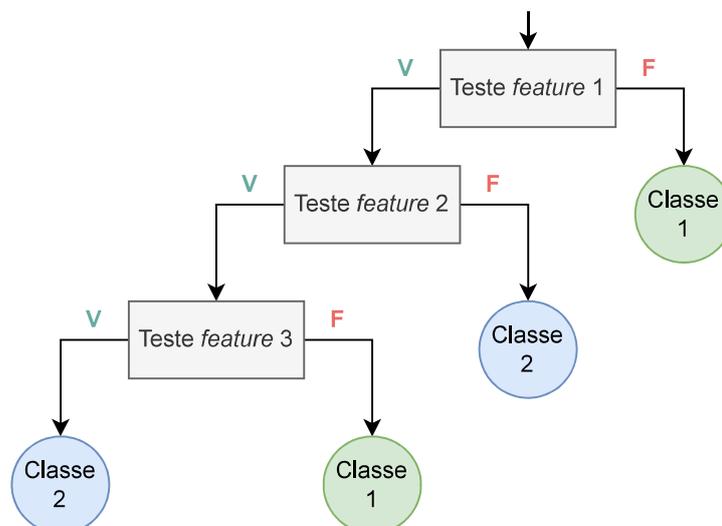
Uma árvore de decisão é um grafo acíclico usado para tomar decisões. Em cada nó, uma expressão envolvendo alguma *feature* é avaliada. Dependendo do resultado da expressão, o nó-ramo à esquerda ou à direita é seguido. Ao chegar a um nó-folha, uma decisão sobre o exemplo é tomada. A Figura 17 exemplifica o uso de uma árvore de decisão em que um exemplo é classificado em Classe 1 ou Classe 2. Os nós-folha são representados por círculos, indicando qual é a classe selecionada. Os retângulos representam os nós-ramo que realizam testes em relação às *features*; dependendo do resultado do teste, outro nó é selecionado. Ao chegar a um nó-folha, a classe predita é definida.

Para treinar uma árvore de decisão, o algoritmo de treinamento começa com apenas um nó. Uma busca entre as *features* é realizada para encontrar uma *feature* e um valor que melhor dividam o *dataset* em dois subconjuntos, de acordo com o valor alvo ou rótulo. Esse processo gera dois nós-folha, um para cada resultado do teste. O procedimento é repetido iterativamente até que alguma limitação seja alcançada, como limite de profundidade ou número de nós (Burkov, 2019).

Para avaliar cada divisão durante o treinamento da árvore de decisão, são utilizados critérios que determinam a qualidade da separação dos dados em cada nó. Entre os critérios mais comuns para classificação estão a Impureza de Gini (*Generalized Inequality Index*) e a Entropia (Suthaharan, 2016).

A Impureza de Gini mede a probabilidade de uma amostra ser classificada incorretamente se for escolhida aleatoriamente, buscando maximizar a homogeneidade dos subconjuntos resultantes. Já a Entropia avalia o grau de desordem ou incerteza nos

Figura 17 – Exemplo de árvore de decisão.



Fonte: Autor.

dados, priorizando divisões que reduzam essa incerteza. Esses critérios ajudam a selecionar a melhor *feature* e o valor de corte que maximizem a separação dos dados, criando subconjuntos mais homogêneos e, conseqüentemente, melhorando a capacidade de predição da árvore de decisão.

A implementação de árvores de decisão da biblioteca *Scikit-learn* (Raschka; Mirjalili, 2019) possui diversos hiperparâmetros que podem ser ajustados para controlar como as árvores de decisão são treinadas. Encontrar um bom ajuste para esses hiperparâmetros é importante para otimizar o desempenho do modelo e evitar problemas como *overfitting* ou *underfitting*. Entre os principais hiperparâmetros que podem ser ajustados estão:

- ***criterion***: Define o critério utilizado para avaliar a qualidade das divisões, podendo ser *gini* ou *entropy*.
- ***min_samples_split***: Especifica o número mínimo de amostras necessárias para que um nó seja dividido.
- ***min_samples_leaf***: Determina o número mínimo de amostras que cada nó-folha deve conter.
- ***max_depth***: Define a profundidade máxima da árvore.
- ***max_leaf_nodes***: Especifica o número máximo de nós-folha que a árvore pode ter.
- ***max_features***: Indica o número máximo de *features* a serem consideradas ao procurar a melhor divisão em cada nó.

Esses hiperparâmetros impõem diversas limitações à árvore treinada. Os parâmetros *max_depth* e *max_leaf_nodes* afetam diretamente o tamanho máximo da árvore. Uma árvore de decisão muito grande pode absorver muitas características específicas do *dataset* de treinamento, prejudicando a capacidade de generalização para dados desconhecidos. No entanto, valores muito pequenos também podem comprometer a qualidade da árvore, levando a um modelo subajustado que não consegue aprender padrões relevantes nos dados, resultando em desempenho insatisfatório tanto no treinamento quanto na validação.

De forma semelhante, *min_samples_split* e *min_samples_leaf* controlam o quão facilmente novos nós são criados na árvore. Se forem usados valores muito baixos, a árvore pode crescer excessivamente, criando divisões baseadas em poucas amostras, o que aumenta o risco de *overfitting*. Por outro lado, valores muito altos podem impedir que a árvore capture divisões importantes, reduzindo sua capacidade de modelar adequadamente os dados. O hiperparâmetro *max_features* limita o número de *features* consideradas em cada divisão. Utilizar um número menor que o total de *features* pode acelerar o processo de treinamento.

Além das árvores de decisão simples, existem métodos baseados em conjuntos de árvores, como *Random Forests* (Parmar; Katariya; Patel, 2018) e *Gradient Boosted Trees* (Ye et al., 2009), que combinam múltiplas árvores para melhorar a robustez e a acurácia dos modelos.

3 TRABALHOS RELACIONADOS

A elevada complexidade computacional dos codificadores de vídeo e, mais recentemente, dos codificadores de nuvens de pontos, tem motivado uma série de pesquisas voltadas ao desenvolvimento de métodos para redução de custo computacional. Neste Capítulo, são apresentados e discutidos os principais trabalhos da literatura que propõem soluções para acelerar o processo de codificação, tanto no contexto de vídeo 2D quanto no contexto específico do V-PCC. O objetivo é contextualizar as abordagens existentes, destacando suas estratégias, resultados e limitações, além de situar as contribuições deste trabalho em relação ao estado da arte.

3.1 Redução de Custo Computacional em Codificadores de Vídeo

Diversos trabalhos têm sido propostos para diminuir o custo computacional dos codificadores de vídeo utilizando técnicas de aprendizado de máquina e métodos heurísticos para acelerar decisões de particionamento e modos de predição. A seguir, são destacados alguns dos principais estudos voltados para o HEVC e o VVC.

Em Correa et al. (2014a), o aprendizado de máquina foi aplicado para reduzir o custo computacional, encerrando precocemente o processo de decisão do tamanho dos blocos das *Coding Tree Units* (CTUs) do HEVC. A configuração temporal RA foi empregada nos experimentos. O modelo proposto em Correa et al. (2014a) atribui os melhores tamanhos de bloco com base nas características do quadro de entrada, evitando testar todos os tamanhos de bloco possíveis (64×64 , 32×32 e 16×16). O trabalho alcançou uma redução média de 37% no custo computacional em comparação com o codificador original, com um aumento marginal no BD-Rate de apenas 0,28%.

Os autores em Li; Xu; Deng (2017) empregam redes neurais convolucionais para reduzir o custo computacional do HEVC, prevendo as decisões das CTUs na configuração temporal *All Intra* (AI). O modelo proposto em Li; Xu; Deng (2017) foi treinado e testado usando um banco de dados em larga escala com padrões diversificados de partição de CTU para cada profundidade de CTU. Essa abordagem reduz o tempo de codificação em 62,25%, com aumentos no BD-Rate de 2,12%.

Em Cho; Kim (2013), é proposto um método rápido de divisão e poda de CUs especificamente para a codificação intra. A abordagem consiste em duas etapas complementares: uma decisão antecipada de divisão da CU e uma decisão antecipada de poda da CU. Ambas as decisões são tomadas em cada nível de profundidade da CU, utilizando uma regra de decisão Bayesiana baseada, respectivamente, em custos de *Rate-Distortion* (RD) de baixa complexidade e custos RD completos. Os parâmetros estatísticos empregados nessas decisões são atualizados periodicamente durante o processo de codificação, permitindo uma adaptação dinâmica às variadas características do sinal de vídeo. Os resultados experimentais indicam que o método proposto consegue reduzir o tempo de codificação do HEVC *Test Model* (HM) em aproximadamente 50%, com um aumento de apenas 0,6% no BD-Rate, para uma configuração de decisão mais conservadora, e uma redução de 63,5% no tempo com 3,5% de aumento no BD-Rate para uma decisão mais agressiva.

No contexto do custo computacional do *Versatile Video Coding* (VVC), Saldanha et al. (2021) apresentam um esquema de redução de custo computacional baseado em aprendizado de máquina para a predição intraquadro. Os autores propõem três soluções para diminuir o custo computacional: uma decisão rápida para os modos Planar/DC, baseada em árvores de decisão, uma decisão rápida para o modo *Matrix Intra Prediction* (MIP), também baseada em árvore de decisão, e uma decisão rápida para o modo *Intra Sub-Partition* (ISP) fundamentada na variância do bloco. Essas estratégias permitem descartar modos de predição improváveis antes do RDO, reduzindo significativamente o tempo de codificação. Os experimentos, realizados com o VVC *Test Model* (VTM) 10.0 sob as condições padrão da JVET, demonstram que o esquema proposto proporciona uma economia média de 18,32% no tempo de codificação, com aumento médio de apenas 0,60% no BD-Rate.

Em Yang et al. (2020), os autores propõem um algoritmo de codificação intra de baixo custo computacional para o padrão VVC. A abordagem é composta por duas estratégias principais: uma decisão de estrutura de partição da CTU de baixo custo computacional baseada em aprendizado estatístico e uma decisão rápida de modo intra utilizando busca por gradiente descendente unidimensional. A primeira estratégia emprega um *framework* de decisão em cascata com classificadores baseados em árvores de decisão para predizer antecipadamente a estrutura de partição *Quadtree with nested Multi-type Tree* (QTMT). A segunda estratégia reduz o número de modos intra candidatos para a estimação de custo RD, selecionando um modo direcional inicial a partir de uma lista de modos mais prováveis e refinando-o por meio de uma busca bidirecional por gradiente descendente. Os resultados mostram uma redução média do custo computacional de codificação de 62,46%, com um BD-Rate de 1,93% em comparação com o VTM 2.0.

3.2 Redução de Custo Computacional no V-PCC

Com o avanço dos padrões de compressão de nuvens de pontos, especialmente o V-PCC, novas estratégias têm sido propostas para mitigar o alto custo computacional associado ao processo de codificação. Os trabalhos podem ser agrupados em duas principais linhas: otimizações dos processos internos do TMC2 e métodos para acelerar a etapa de codificação de vídeo dos subfluxos do V-PCC.

Os autores em Kim; Kim (2021) propõem um método para o refinamento de segmentação baseado em grade (*Grid-based Refining Segmentation*), focado na etapa de geração de *patches*. O método introduz um esquema de classificação de voxels, diferenciando-os em *direct edge-voxels*, *indirect edge-voxels* e *no edge-voxels*, com base em índices de uniformidade da distribuição de planos de projeção. Os experimentos realizados com o TMC2 12.0 mostram que a abordagem proposta reduz, em média, 17,1% do tempo total do TMC2 (desconsiderando as etapas de codificação de vídeo) e, especificamente nos passos de refinamento, as reduções de tempo foram de 60,7% na configuração RA e 62,5% na configuração AI. Em relação ao impacto na eficiência de codificação, os valores médios de BD-Rate observados foram de 0,12% para D1 e 0,14% para D2 (geometria), e de 0,06%, -0,03% e -0,27% para os atributos Luma, Cb e Cr, respectivamente.

Em Seidel et al. (2021), são propostas modificações no método de refinamento de segmentação do TMC2, com foco na otimização do acesso à memória. O trabalho identifica que, mesmo após a adoção do *Grid-based Refining Segmentation*, essa etapa ainda representa mais de 50% do tempo de execução do TMC2 em diversas sequências, devido ao padrão de acesso disperso à memória e ao uso intensivo de estruturas como *hash maps*. Duas soluções são apresentadas: a primeira reduz alocações desnecessárias de memória durante a busca por vizinhos, enquanto a segunda reorganiza o acesso aos dados para aumentar a localidade de referência, utilizando pré-processamento e estruturas auxiliares para agrupar e ordenar os acessos. Os resultados demonstram que as modificações propostas podem reduzir o tempo de execução do TMC2 em até 46,31%, sem impacto na eficiência de compressão, já que o *bitstream* gerado permanece idêntico ao do TMC2 original.

Em Jia et al. (2024) é proposto algoritmos simples baseados em grade para o refinamento da segmentação. O método parte do princípio de que, ao classificar voxels conforme a uniformidade da distribuição dos índices de projeção, é possível omitir etapas de refinamento para regiões uniformes, reduzindo o número de operações necessárias. Os autores introduzem dois algoritmos: o primeiro (p1) verifica, a cada iteração, se houve alteração na distribuição dos índices de projeção dos voxels e, caso não haja, omite o processamento desses elementos; o segundo (p2) interrompe o refinamento quando a proporção de voxels alterados atinge um limiar mínimo. Os

experimentos, realizados com o *software* de referência TMC2 v15.0, demonstram que os algoritmos p1 e p2 proporcionam reduções de até 79,21% e 79,53% no tempo de segmentação de refinamento. Além disso, observou-se melhora na eficiência de compressão, com ganhos de BD-Rate entre 0,1% e 0,9% usando o modo AI.

No que diz respeito à etapa de codificação de vídeo dos subfluxos do V-PCC, diferentes abordagens têm sido propostas para acelerar o processo, seja por meio de decisões rápidas de particionamento, uso de aprendizado de máquina ou estratégias de busca adaptativa. Os principais resultados da literatura incluem reduções expressivas no tempo de codificação, geralmente com impactos mínimos ou controlados na eficiência de compressão.

Em Gao et al. (2023), os autores propõem um algoritmo de baixa complexidade para acelerar a codificação intra de nuvens de pontos no padrão V-PCC. A abordagem explora as fortes correlações entre as sequências 2D projetadas (ocupação, geometria e atributos) a partir de quadros 3D, introduzindo informações de projeção cruzada para melhorar a previsão de partições de CUs. Além disso, é desenvolvido um método de aprendizado orientado por taxa-distorção para aumentar a precisão das decisões de partição das CUs, considerando as perdas de codificação geradas por decisões incorretas. O *framework* proposto inclui extração de características direcionadas e decisões de partição para codificação intra e inter das sequências de geometria e atributos. Essa abordagem, utilizando o modo AI, reduziu o tempo total de compressão dos subfluxos em 57,80%, com BD-Rate de 0,08% e 0,33% de Geom.BD-TotalRate (D1 e D2) e 0,16% para o canal Luma, considerando Attr.BD-TotalRate.

Em Li et al. (2024), é proposto um método de decisão rápida do tamanho das CUs para subfluxos de geometria no padrão V-PCC, utilizando aprendizado não supervisionado. A abordagem baseia-se em um algoritmo de *clustering* hierárquico, projetado para lidar com as características específicas dos subfluxos de geometria, que apresentam regiões com bordas acentuadas e escalas variadas. Para melhorar a precisão na determinação do tamanho ideal das CUs, o método introduz um limiar de ligação adaptativo no processo de *clustering*, ajustando-se conforme diferentes parâmetros de quantização e tamanhos de CU. Sob a configuração AI do V-PCC, o método proposto reduziu o tempo de codificação dos subfluxos de geometria em média entre 56,7% e 69,3%. Essa redução significativa no custo computacional de codificação ocorre com um aumento mínimo na taxa de *bits*, variando de apenas 0,1% a 0,5% no BD-TotalRate.

Em Lin et al. (2021), os autores propõem um algoritmo para acelerar a decisão de particionamento das CUs no contexto do V-PCC utilizando o codificador HEVC. A abordagem introduz o *Blocky Occupancy Flag* (BOF), que utiliza mapas de ocupação para indicar, em nível de bloco, áreas relevantes para codificação. Esse sinalizador permite a terminação antecipada do processo de RDO das CUs, evitando processamento

desnecessário em regiões não ocupadas. Além disso, o algoritmo incorpora informações de homogeneidade espacial 2D/3D derivadas das imagens de geometria para aprimorar a decisão de particionamento. Uma modificação no RDO também é proposta, considerando a estrutura de *Picture Order Count* (POC) do HEVC/V-PCC. O método proposto alcançou uma redução de cerca de 55% no tempo gasto pelo HEVC no V-PCC, com resultados de BD-Rate de 0,67% e 0,82% para D1 e D2, considerando Geom.BD-GeomRate, e -2,49% para atributos, considerando Attr.BD-TotalRate.

O estudo em Wang et al. (2024) introduz uma abordagem de aprendizado de máquina para encerramento precoce visando a redução do custo computacional na codificação de mapas de geometria e atributos no nível de CU. Em vez de usar o codificador de vídeo HEVC, os autores optaram pelo padrão VVC (Bross et al., 2021), devido à sua capacidade de manter a qualidade subjetiva da imagem, embora com um custo computacional até dezenove vezes maior que o HEVC na configuração AI. O trabalho alcançou aproximadamente 55% de redução no tempo de codificação no V-PCC ao usar o VVC modificado como codificador de vídeo, em comparação com o uso do VVC sem o método proposto. O impacto na eficiência de codificação foi de 0,4% para D1 e 0,56% para D2, considerando Geom.BD-GeomRate. Para os atributos, os valores foram 0,17% para Luma, e 0,61% e -1,06% para Cb e Cr, considerando Attr.BD-TotalRate e utilizando o modo AI do V-PCC.

Em Song; Liu; Zhang (2025), é proposto um método rápido de particionamento adaptativo das CUs que integra informações de ocupação, redes neurais convolucionais (*Convolutional Neural Network* – CNN) e otimização Bayesiana. Inicialmente, as CUs são classificadas em regiões densas, esparsas e compostas a partir do mapa de ocupação, sendo a decisão inicial realizada por uma CNN. Para regiões com baixa confiança na classificação, a otimização Bayesiana é empregada para refinar o particionamento e aumentar a precisão. Os resultados experimentais demonstram uma redução média de 57,37% no tempo de codificação do subfluxo de geometria e 54,43% para o subfluxo de atributos, com aumento marginal no BD-Rate de 0,42% para D1, 0,56% para D2, 0,18% para Luma, 0,68% para Cb e redução de 1,04% para Cr, considerando Geom.BD-GeomRate e Attr.BD-TotalRate, respectivamente. Não é especificado qual configuração temporal foi utilizada.

Em Que; Li (2023), é apresentado um método de decisão rápida de modos de codificação baseado em uma *Portable Perceptron Network* (PPN), visando acelerar o processo de compressão do V-PCC sob a configuração RA. O método propõe a extração de características simples, alimentando uma rede neural leve com 1,27K de parâmetros. Uma função de perda adaptativa é utilizada, que pondera os erros conforme o custo de RD. Os resultados demonstram uma redução média de 43,13% no tempo de codificação total, com resultados médios de BD-Rate de -0,4% para D1 e D2, e 0,1%, -0,8% e -0,2% para os canais Y, Cb e Cr, respectivamente.

De modo geral, observa-se que as soluções para redução de custo computacional em codificadores de vídeo e no V-PCC têm evoluído para o uso de técnicas baseadas em aprendizado de máquina, decisões rápidas e otimizações específicas para cada etapa do processo de codificação. Os resultados reportados na literatura evidenciam reduções expressivas no tempo de codificação, com impactos controlados na eficiência de compressão, especialmente quando os métodos são adaptados ao contexto e às características dos dados processados.

A maioria dos trabalhos focados no V-PCC encontrados avaliam ou propõem soluções apenas para o modo AI do V-PCC. Além disso, observa-se que poucos trabalhos realizam uma análise detalhada do impacto de soluções originalmente desenvolvidas para vídeos 2D convencionais quando aplicadas ao contexto de compressão de nuvens de pontos dinâmicas.

Neste trabalho, é realizada uma avaliação da adaptação de um esquema de redução de custo computacional para vídeos 2D no V-PCC, analisando seus efeitos tanto em termos de tempo de codificação quanto de eficiência de compressão. Esta análise permite identificar de forma sistemática as limitações e potencialidades da transferência de soluções entre domínios distintos, contribuindo para um entendimento mais aprofundado sobre a aplicabilidade dessas técnicas no contexto do V-PCC. Além do mais, as soluções propostas neste trabalho abrangem tanto o modo RA quanto o modo AI do V-PCC, ampliando o escopo de avaliação em relação à maioria dos trabalhos da literatura, que se restringem principalmente ao modo AI. No próximo capítulo, são apresentadas as soluções propostas neste trabalho, destacando suas diferenças em relação às abordagens discutidas e os resultados obtidos em cenários práticos.

4 ESQUEMA PARA REDUÇÃO DO CUSTO COMPUTACIONAL DE CODIFICAÇÃO DE VÍDEOS 2D CONVENCIONAIS NO V-PCC

Este capítulo tem como objetivo testar a aplicação de um esquema para redução do custo computacional do *software* de referência do codificador de vídeo HEVC, originalmente desenvolvido para emprego na codificação de vídeos 2D convencionais, no contexto da codificação de nuvens de pontos dinâmicas com o *software* de referência do V-PCC, especificamente na etapa de codificação de vídeo dos subfluxos de vídeo, onde o HEVC é utilizado. Para isso, uma solução existente na literatura (Correa et al., 2014a) foi adaptada para funcionar no TMC2. Este capítulo apresenta a metodologia utilizada para o desenvolvimento dessa solução, bem como os resultados obtidos em relação à redução do tempo de codificação e ao impacto na eficiência de codificação.

4.1 Solução para Redução do Custo Computacional do HEVC

A fim de testar o uso de soluções para redução do custo computacional do HEVC em vídeos convencionais, foi utilizada a solução proposta por Correa et al. (2014a). Essa solução faz uso de árvores de decisão para decidir o particionamento dos blocos das CTUs previstas com o modo inter-quadros no HEVC (Modo RA). Uma árvore de decisão treinada com dados extraídos do processo de codificação é utilizada em cada profundidade da CTU onde é possível realizar uma subdivisão, ou seja, nas profundidades 64×64 , 32×32 e 16×16 . Cada árvore recebe *features* sobre a CU sendo avaliada e, a partir dos valores obtidos, a CU é classificada como *skip* ou *não-skip*. Se uma CU é classificada como *não-skip*, as CUs da profundidade inferior são avaliadas. Caso seja classificada como *skip*, o processo de busca de particionamento para aquele ramo da CTU é interrompido, significando que aquela profundidade de CU foi considerada uma boa candidata para ser selecionada pelo processo de RDO, não sendo necessário testar os níveis inferiores.

Ao interromper o processo de RDO antecipadamente, nem todas as possibilidades de particionamento são testadas, reduzindo assim o tempo de codificação. Caso

a classificação esteja correta, ou seja, aquela CU apresenta os melhores resultados de taxa-distorção (*rate-distortion*, RD), haverá ganho de tempo de codificação sem nenhuma perda em termos de eficiência de codificação. Se for selecionada uma CU em uma profundidade inferior àquela com os melhores resultados de RD, não haverá perdas de eficiência de codificação, podendo haver ganho de tempo de codificação se a busca for interrompida antes do último nível da CTU. Nesse caso, não há perdas de eficiência, pois, embora o processo de busca não tenha sido interrompido na profundidade da CU com os melhores resultados, esta foi avaliada e o processo de RDO ainda selecionará essa profundidade como a melhor opção. Por outro lado, se for selecionada uma CU em uma profundidade superior àquela com os melhores resultados de RD, haverá ganhos de tempo de codificação, uma vez que a busca é interrompida. Porém, também haverá perdas de eficiência de codificação, já que a CU com os melhores resultados não foi avaliada e, portanto, não poderá ser selecionada pelo processo de RDO. Nesses casos, um particionamento subótimo é selecionado.

Para o treinamento das árvores de decisão, foram utilizados dados coletados durante a codificação de sequências de teste, abrangendo diferentes resoluções espaciais e temporais. As *features* utilizadas estão listadas abaixo. A seleção dessas variáveis foi baseada na métrica de ganho de informação, e o algoritmo C4.5 foi utilizado para a construção das árvores, sendo uma para cada nível de CU: 64×64 , 32×32 e 16×16 .

- **Neighbor Depth:** Média de profundidade utilizada nas CTUs vizinhas já codificadas (CTUs acima, à esquerda, acima à esquerda, acima à direita e CTUs colocalizadas nos primeiros quadros das duas listas de referência do HEVC).
- **Partition:** Particionamento da PU selecionada para a CU sendo avaliada ($2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$ ou $nR \times 2N$).
- **Ratio($2N \times 2N$, MSM):** Razão entre os custos de RD da PU $2N \times 2N$ e MSM.
- **RD($2N \times 2N$), RD($2N \times N$) e RD(MSM):** Custo de RD dos modos de PU $2N \times 2N$, $2N \times N$ e MSM, respectivamente.
- **NormDiff($2N \times 2N$, MSM):** Diferença normalizada entre os custos de RD dos modos $2N \times 2N$ e MSM. Calculada conforme a Equação 6
- **MergeFlag e SkipMergeFlag:** *Flags* indicando se a CU foi predita com o modo *skip* ou MSM.

$$NormDiff(a, b) = \left| \frac{RD(a) - RD(b)}{RD(b)} \right| \quad (6)$$

Ao serem integradas ao codificador HM 12 e testadas com a CTC do HEVC, os resultados experimentais demonstraram uma redução média de 37% no tempo total de codificação, com um aumento médio de 0,28% na BD-Rate, conforme reportado em Correa et al. (2014a). Esses resultados representam uma boa relação entre custo computacional e eficiência de codificação.

4.2 Metodologia

A solução de Correa et al. (2014a) foi adaptada para funcionar com o HM versão 16.20, utilizada no TMC2 versão 22.1. Por se tratar de uma versão diferente daquela empregada em Correa et al. (2014a) (versão 12), adaptações precisaram ser feitas no *software* de referência para implementar a solução em uma versão mais recente do HM.

As mesmas árvores de decisão utilizadas em Correa et al. (2014a) foram empregadas na versão adaptada para funcionamento no V-PCC. As mesmas *features* também foram mantidas. No entanto, a *feature Neighbor Depth* foi modificada, pois, devido a mudanças no *software* de referência do HEVC, não é possível acessar diretamente as informações de profundidade das CTUs colocalizadas. Assim, a *feature Neighbor Depth* passou a considerar a profundidade média das CTUs vizinhas já codificadas do mesmo quadro (CTUs acima, à esquerda, acima à esquerda e acima à direita). Por se tratar de uma média de várias CTUs, o impacto dessa mudança é minimizado.

Essa solução foi aplicada na codificação dos subfluxos de geometria e atributo. Optou-se por não utilizar as árvores de decisão na codificação do mapa de ocupação devido à pequena quantidade de tempo gasto na codificação desse tipo de subfluxo (conforme apresentado na Tabela 2), minimizando assim o possível impacto no tempo total de codificação do V-PCC. Além disso, os impactos em termos de eficiência de codificação poderiam ser grandes o suficiente para inviabilizar os ganhos de tempo nesse subfluxo. Uma nova versão do *software* de referência do V-PCC foi criada, com as modificações do HM 16.20, para utilizar as árvores de decisão no processo de codificação das CTUs.

4.3 Resultados Experimentais

Para avaliar o impacto da solução, foram realizados testes utilizando as sequências de teste de 10 *bits* especificadas no CTC do V-PCC (MPEG, 2020), sendo elas: *Longdress*, *Loot*, *Queen*, *Redandblack* e *Soldier*. Os primeiros 64 quadros de cada sequência de teste foram codificados tanto com a implementação padrão do TMC2 versão 22.1 quanto com a versão modificada, que utiliza os modelos de árvores de decisão. Cada sequência foi codificada com as cinco configurações de bitrate do V-PCC

(r1, r2, r3, r4 e r5), utilizando a configuração temporal *random access*. Os experimentos foram realizados em um computador com processador AMD Opteron 8276s e 128 GB de memória RAM. Cuidados foram tomados quanto ao número de experimentos em paralelo e à quantidade de RAM utilizada. Todos os experimentos foram realizados com a mesma quantidade de codificações em paralelo.

Os resultados das codificações foram avaliados utilizando o *software Distortion Metric* para o cálculo das métricas de qualidade objetiva das nuvens de pontos reconstruídas (ponto a ponto (D1) e ponto a plano (D2) para geometria, e PSNR dos canais de Luminância (Luma), Crominância azul (Cb) e Crominância vermelha (Cr)), conforme especificado no CTC (MPEG, 2020). Os resultados do tempo total de codificação e de cada subfluxo de vídeo também foram coletados.

Os resultados médios de redução de tempo para cada sequência de teste utilizada são apresentados na Tabela 3. São exibidos os valores de redução do tempo de codificação dos subfluxos de geometria e atributo, além da redução total, que considera todo o tempo utilizado para codificar os três subfluxos no HEVC, conforme apresentados nas Equações 7 e 8, onde a e b são as codificações com o *software* original e modificado, respectivamente. Os resultados mostram que a solução foi efetiva em reduzir o tempo de codificação dos subfluxos, com uma redução média de 54,72% nos subfluxos de atributo, chegando ao máximo de 65,37% na sequência *loot*. Para os subfluxos de geometria, a redução média do tempo de codificação foi de 42,96%, com redução máxima de 55,4% na sequência *queen*. Esses resultados indicam que os modelos de aprendizado de máquina conseguiram alcançar uma boa redução do custo computacional em ambos os subfluxos. Foram observadas reduções de tempo de codificação maiores nos subfluxos de atributo, o que pode ser explicado pelo fato de esse tipo de subfluxo apresentar características visuais mais próximas ao conteúdo de vídeo 2D usual (imagens coloridas, texturas da nuvem de pontos), sobre o qual os modelos foram inicialmente treinados, enquanto os subfluxos de geometria apresentam características mais distintas (imagens em escala de cinza, mapeando a profundidade dos *patches*).

$$TempoTotal(a) = TempoOcup(a) + TempoGeom(a) + TempoAtrib(a) \quad (7)$$

$$ReduoTotal(a, b) = \frac{TempoTotal(b)}{TempoTotal(a)} - 1 \quad (8)$$

No entanto, também é necessário avaliar o impacto da solução em relação à eficiência de codificação. Para isso, foram calculados os valores de BD-Rate sobre as métricas de geometria e atributos para cada sequência de teste utilizada. Os resultados são apresentados nas Tabelas 4 e 5. O BD-Rate foi calculado considerando os valores de *bitrate* do subfluxo (Attr.BD-AttrRate e Geom.BD-GeomRate) e também considerando o valor de *bitrate* final do *bitstream* gerado pelo V-PCC (Attr.BD-TotalRate e

Tabela 3 – Resultados médios de redução de tempo de codificação dos subfluxos de geometria e atributo para a esquema adaptado de vídeos 2D.

Sequência de Teste	Red. Tempo		
	Total	Geometria	Atributo
longdress	-38,67%	-34,74%	-43,02%
loot	-48,79%	-40,06%	-65,37%
queen	-57,79%	-55,40%	-59,26%
redandblack	-39,02%	-34,45%	-46,31%
soldier	-56,66%	-49,03%	-62,56%
Média	-48,19%	-42,96%	-54,72%

Geom.BD-TotalRate).

A Tabela 4 mostra os resultados de BD-Rate relacionados aos atributos. Os resultados indicam um valor médio de Attr.BD-AttrRate de 3,89% para o canal de luminância, evidenciando uma perda de eficiência de codificação para este componente quando se considera apenas o *bitrate* do subfluxo de atributos. Os canais de crominância (Cb e Cr) apresentam impacto muito menor, com médias de 0,05% e 1,59%, respectivamente. O valor próximo de zero para Cb sugere que, em média, a eficiência de codificação para este canal é minimamente afetada. Quando se considera o *bitrate* total, o impacto no BD-Rate é maior em todos os canais: 6,25%, 3,43% e 4,52% para os canais de Luma, Cb e Cr, respectivamente. O aumento percentual do BD-Rate é consistentemente maior no Attr.BD-TotalRate em comparação com o Attr.BD-AttrRate para todos os canais, o que é esperado, pois o *TotalRate* reflete o impacto global, considerando também o *bitrate* gasto pelo subfluxo de geometria.

A sequência *queen* apresenta os maiores aumentos de BD-Rate em quase todos os canais, especialmente para Luma (8,58% em Attr.BD-AttrRate e 13,27% em Attr.BD-TotalRate). Isso pode indicar que as características específicas dessa sequência tornam a otimização proposta menos eficaz ou até prejudicial para a eficiência de codificação dos atributos. Como essa sequência é a única composta por conteúdo sintético entre as sequências utilizadas, e considerando que o *dataset* usado para treinar os modelos em Correa et al. (2014a) é formado por conteúdo captado em ambientes reais, é esperado que os modelos apresentem melhor desempenho em conteúdos mais próximos do *dataset* de treinamento. Em contraste, as sequências *loot* e *soldier* apresentam valores negativos de BD-Rate para os canais de crominância Cb e Cr no Attr.BD-AttrRate (-0,46% e -0,75% para *loot*; -4,40% e -1,49% para *soldier*). Isso sugere que, para essas sequências e canais específicos, a abordagem pode ter proporcionado uma codificação mais eficiente dos atributos em seu próprio subfluxo. No entanto, esse ganho não se traduz necessariamente em melhorias no Attr.BD-TotalRate, onde os valores para Cb e Cr ainda permanecem positivos.

Os ganhos de eficiência podem parecer contraditórios, considerando que a implementação padrão do HEVC testa todas as possibilidades de particionamento e se-

leciona a melhor. Assim, os modelos ou selecionarão o mesmo particionamento da implementação padrão (escolhido pelo processo de RDO, ou seja, com melhor taxa de RD), ou um particionamento subótimo, com uma taxa de RD inferior àquela selecionada pelo processo de RDO na implementação padrão. No entanto, é importante destacar que a qualidade das nuvens de pontos é avaliada a partir das nuvens de pontos reconstruídas, e não com base na qualidade objetiva dos subfluxos de vídeo. Além disso, existem regiões nos quadros planejados sem informação de pontos válidos (espaço entre os *patches*), e perdas de qualidade nessas regiões não afetam a qualidade da nuvem reconstruída. Ademais, o V-PCC faz uso de dois (padrão) ou mais *layers*, chamados *far-layer* e *near-layer*, utilizados para armazenar pontos que possam ter sido projetados para a mesma coordenada no *patch* pelo processo de planificação. No entanto, em regiões onde não há pontos duplicados, a informação é apenas copiada do *near-layer*, e perdas nessas regiões também não se refletem na nuvem de pontos reconstruída. Isso também pode justificar a inconsistência do impacto do BD-Rate entre as sequências de teste.

Tabela 4 – Resultados de BD-Rate de Atributo para o esquema adaptado.

Sequência de Teste	Attr.BD-AttrRate			Attr.BD-TotalRate		
	Luma	Cb	Cr	Luma	Cb	Cr
longdress	2,35%	3,04%	3,58%	3,94%	3,66%	4,04%
loot	1,26%	-0,46%	-0,75%	3,85%	2,31%	3,19%
queen	8,58%	0,64%	4,11%	13,27%	6,06%	8,57%
redandblack	2,25%	1,75%	1,92%	4,62%	3,97%	4,09%
soldier	1,61%	-4,40%	-1,49%	5,58%	1,13%	2,70%
Média	3,89%	0,05%	1,59%	6,25%	3,43%	4,52%

Os resultados referentes às métricas relacionadas à geometria são apresentados na Tabela 5. Os resultados médios foram de 4,23% para D1 e 4,61% para D2 ao considerar o Geom.BD-GeomRate, indicando uma razoável perda de eficiência de codificação nos subfluxos de geometria. Esses valores também são observados nas métricas de Geom.BD-TotalRate, com 4,83% para D1 e 4,99% para D2. Isso indica que, ao usar a solução, o processo de RDO está sendo interrompido antes da melhor profundidade ser avaliada na maioria dos casos. Houve uma perda de eficiência de codificação quase uniforme em todas as sequências de teste. Considerando o Geom.BD-GeomRate, a sequência *soldier* apresentou os menores impactos de BD-Rate (2,40% para D1 e 2,82% para D2), enquanto *redandblack* apresentou as maiores perdas (5,53% para D1 e 6,17% para D2). No Geom.BD-TotalRate, *queen* exibiu o maior aumento (6,94% para D1 e 6,53% para D2), e *loot* o menor (3,65% para D1 e 3,77% para D2). É importante destacar que o Geom.BD-TotalRate também é influenciado pelo subfluxo de atributo, o que pode justificar os valores elevados de Geom.BD-TotalRate da sequência *queen*, que obteve altos valores de BD-Rate nos

atributos.

Tabela 5 – Resultados de BD-Rate de Geometria para o esquema adaptado.

Sequência de Teste	Geom.BD-GeomRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
longdress	4,63%	5,61%	3,98%	4,69%
loot	4,54%	4,96%	3,65%	3,77%
queen	3,91%	4,19%	6,94%	6,53%
redandblack	5,53%	6,17%	4,45%	4,78%
soldier	2,40%	2,82%	5,12%	5,17%
Média	4,23%	4,63%	4,83%	4,99%

No geral, a solução apresentou melhor desempenho na codificação dos subfluxos de atributo, o que já era esperado, pois os modelos foram treinados com vídeo 2D convencional. Os subfluxos de geometria possuem características distintas do conteúdo de vídeo 2D convencional, o que pode levar a uma generalização menos eficaz dos modelos para esse tipo de dado, contribuindo para as perdas de BD-Rate. A adaptação da *feature Neighbor Depth* (removendo o acesso às CTUs colocalizadas) também pode ter impacto, já que as árvores foram treinadas considerando esses valores na média das CTUs vizinhas.

A Tabela 6 apresenta a razão ou *trade-off* de BD-Rate por tempo de codificação reduzido, considerando as métricas Geom.BD-GeomRate e Attr.BD-AttrRate, e os tempos de codificação de cada subfluxo. Esses valores indicam qual a perda (ou ganho, para valores negativos) de eficiência de codificação por cada ponto percentual de tempo de codificação reduzido. Valores menores indicam uma redução de custo computacional com menor impacto na eficiência de codificação. Os resultados médios mostram que, para subfluxos de geometria, a razão é de 0,11 para D1 e 0,12 para D2. Para subfluxos de atributo, os valores obtidos foram 0,06 para Luma, 0,01 para Cb e 0,03 para Cr. Esses valores mostram que a solução utilizada reduziu o custo computacional de forma mais eficiente para os fluxos de atributo. Comparando com os resultados obtidos em Correa et al. (2014a) para vídeos 2D convencionais (valores médios de 37% de redução de tempo e aumento de BD-Rate de 0,28%), a razão de BD-Rate por Redução de Tempo (0,007) é mais próxima dos valores obtidos para subfluxos de atributo.

Cabe destacar que, apesar do impacto na eficiência de codificação, a abordagem utiliza praticamente metade do tempo de codificação da implementação padrão do TMC2 22.1 (redução de 48,19% no tempo total gasto pelo HEVC) e, em algumas aplicações, esse *trade-off* é aceitável, como em aplicações em tempo real ou em sistemas com recursos limitados, como bateria e poder de processamento. No entanto, os resultados indicam que soluções especializadas podem apresentar desempenho superior em relação a soluções adaptadas do contexto de vídeo 2D, principalmente

para a codificação dos subfluxos de geometria.

Tabela 6 – Razão de BD-Rate por redução de tempo de codificação para o esquema adaptado.

Sequência	BD-Rate / Red. Tempo				
	Geom.BD-GeomRate		Attr.BD-AttrRate		
	D1	D2	Luma	Cb	Cr
longdress	0,133	0,162	0,055	0,071	0,083
loot	0,113	0,124	0,019	-0,007	-0,011
queen	0,071	0,076	0,145	0,011	0,069
redandblack	0,160	0,179	0,049	0,038	0,041
soldier	0,049	0,057	0,026	-0,070	-0,024
Média	0,105	0,120	0,059	0,008	0,032

5 SOLUÇÃO PARA REDUÇÃO DO CUSTO COMPUTACIONAL DO V-PCC

A abordagem adaptada para o V-PCC, originalmente proposta para vídeos 2D convencionais e apresentada no capítulo anterior, apresentou bons resultados na redução de custo computacional. No entanto, os resultados de BD-Rate indicam que soluções especializadas podem alcançar uma melhor relação entre BD-Rate e redução de custo computacional. Isso é evidenciado pela diferença dos resultados apresentados na Tabela 6, em que os valores obtidos para subfluxos de atributos (mais similares ao conteúdo usado para treinamento) foram melhores do que os de subfluxos de geometria. Isso demonstra que os modelos treinados com vídeos 2D de Correa et al. (2014a) podem não ser os mais adequados para subfluxos de geometria.

Neste capítulo, é apresentada uma solução especializada para a redução de custo computacional do V-PCC, baseada em modelos de aprendizado de máquina treinados com dados extraídos do próprio V-PCC. Essa solução combina duas abordagens para redução de custo computacional, aplicadas à etapa de codificação dos subfluxos de geometria e atributo do V-PCC. A primeira abordagem consiste em uma adaptação direta da solução apresentada no Capítulo 4, utilizando árvores de decisão para decidir o particionamento das CTUs, com modelos treinados especificamente para o V-PCC. A segunda abordagem emprega modelos de aprendizado de máquina para realizar a interrupção antecipada do processo de escolha do particionamento de PUs, também treinados com dados do V-PCC. Essas duas abordagens foram combinadas para potencializar a redução do custo computacional do V-PCC, atuando nos níveis de CTU e PU, respectivamente.

A metodologia utilizada para o desenvolvimento dessa solução é apresentada neste capítulo, assim como os resultados experimentais obtidos. Os resultados intermediários de cada abordagem individual são apresentados e discutidos, bem como os resultados finais obtidos com a combinação das duas abordagens.

5.1 Metodologia

A fim de treinar os modelos de aprendizado de máquina utilizados na solução proposta, foram coletados dados do processo de codificação do V-PCC. Esses dados foram extraídos do TMC2 versão 22.1 (MPEG, 2024), que utiliza o HM 16.20 como codificador de vídeo. Modificações foram realizadas no HM para coletar as informações necessárias ao treinamento dos modelos de aprendizado de máquina para cada abordagem.

Para criar o *dataset* de treinamento de cada abordagem, foram utilizadas as sequências *BlueBackpack*, *BlueSpin*, *BlueSquat*, *CasualSpin*, *CasualSquat*, *ElegantDance*, *ElegantWave*, *FlowerDance*, *FlowerWave* e *Gymnast* (Gautier et al., 2023), com precisão de 11 *bits*. Também foram realizados testes usando essas mesmas sequências, mas com precisão de 10 *bits*. Porém, os modelos treinados com essas sequências não apresentaram resultados tão bons quanto os treinados com as sequências de 11 *bits*. Também foram realizados testes com ambas as versões de precisão (10 e 11 *bits*) para as sequências de teste listadas, mas, novamente, os modelos treinados apresentaram dificuldades em generalizar para as sequências de teste. A partir disso, concluiu-se que as sequências com precisão de 10 *bits* possuíam alguma característica que dificultava a generalização dos modelos treinados e, portanto, optou-se por utilizar apenas as sequências com precisão de 11 *bits* para o treinamento dos modelos.

Os primeiros 64 quadros dessas sequências foram codificados com o TMC2 versão 22.1 (MPEG, 2024), utilizando a configuração temporal RA e as cinco configurações de taxa do V-PCC (r1, r2, r3, r4 e r5), conforme descrito nas CTC (MPEG, 2020) do V-PCC.

5.1.1 Abordagem de Interrupção Antecipada de CTUs

Para a abordagem de interrupção antecipada de CTUs, foram coletadas informações sobre as CTUs de profundidade 64, 32 e 16, para os subfluxos de geometria e atributos. Esses dados foram organizados em seis *datasets* diferentes, de acordo com o subfluxo e a profundidade da CTU: *Geometria_64x64*, *Geometria_32x32*, *Geometria_16x16*, *Atributo_64x64*, *Atributo_32x32* e *Atributo_16x16*. As informações coletadas foram as mesmas utilizadas no esquema apresentado no Capítulo 4. Contudo, como já discutido, não foi possível acessar as informações de profundidade das CTUs colocalizadas. Por esse motivo, a *feature Neighbor Depth* foi adaptada para desconsiderar as CTUs colocalizadas. Assim, a média das profundidades das CTUs vizinhas passou a ser calculada considerando apenas as CTUs do mesmo quadro, ou seja, as CTUs acima-esquerda, acima, acima-direita e esquerda. As demais *features* permaneceram inalteradas. Adicionalmente, foi coletada uma variável alvo para indicar se a

CTU sendo testada apresentou o melhor resultado de RDO (valor 1) ou não (valor 0).

Como a maioria dos exemplos coletados pertence à classe 0 (ou seja, CTUs não selecionadas pelo processo de RDO), foi necessário realizar um balanceamento dos dados para cada *dataset*. Para isso, foi verificado o número de exemplos da classe minoritária (CTUs selecionadas) para cada sequência e taxa utilizada. O menor valor encontrado foi utilizado para subamostrar as demais sequências e taxas. Dessa forma, o número de exemplos da classe 1 (CTUs selecionadas) e da classe 0 (CTUs não selecionadas) ficou igual para cada sequência e taxa do *dataset*. Esse processo foi realizado de forma independente para cada um dos seis *datasets* mencionados anteriormente.

Após isso, foram treinados modelos de aprendizado de máquina utilizando o algoritmo de árvore de decisão da biblioteca Scikit-learn (Pedregosa et al., 2011). Um processo de busca de hiperparâmetros foi realizado para encontrar os melhores parâmetros para cada um dos modelos. A busca foi conduzida utilizando o algoritmo de busca aleatória (*RandomizedSearchCV*) da biblioteca Scikit-learn (Pedregosa et al., 2011). O espaço de busca utilizado está apresentado na Tabela 7. O número total de combinações de hiperparâmetros foi de 286.000, e o número de combinações testadas foi de 2.860 (1% do total de combinações). A validação cruzada foi realizada utilizando 5 *folds* e a métrica de avaliação utilizada foi o F1-Score.

Tabela 7 – Espaço de busca de hiperparâmetros para o ajuste dos modelos de árvores de decisão.

Hiperparâmetro	Espaço de Busca	Tamanho
a. Criterion	{'gini', 'entropy'}	2
b. min_samples_split	[10–90 (passo 20)] \cup [100–1150 (passo 250)]	10
c. min_samples_leaf	[10–90 (passo 20)] \cup [100–1150 (passo 250)]	10
d. max_depth	[2–14 (passo 1)]	13
e. max_leaf_nodes	[10–90 (passo 20)] \cup [100–900 (passo 200)]	10
f. max_features	[1–11 (passo 1)]	11
Total		286,000

Os seis ajustes de hiperparâmetros selecionados são apresentados na Tabela 8. As colunas a, b, c, d, e e f correspondem aos hiperparâmetros apresentados na Tabela 7, ou seja, *criterion*, *min_samples_split*, *min_samples_leaf*, *max_depth*, *max_leaf_nodes* e *max_features*. A última coluna apresenta o F1-Score obtido para cada um dos modelos. Os resultados mostram que os modelos de aprendizado de máquina conseguiram alcançar bons resultados, com F1-Scores superiores a 0,80 para todos os modelos. Os melhores resultados foram obtidos para os subfluxos de atributos, com F1-Scores superiores a 0,90. Os modelos do subfluxo de geometria apresentaram resultados levemente inferiores, com F1-Scores entre 0,81 e 0,85. Esses resultados indicam que os modelos de aprendizado de máquina conseguiram aprender as características das CTUs selecionadas pelo processo de RDO, mesmo sem

acesso às CTUs colocalizadas.

Tabela 8 – Resultados da busca aleatória para cada *dataset*.

Dataset	Hiperparâmetro (conforme Tabela 7)						F1-Score
	a	b	c	d	e	f	
Geometria_16x16	gini	50	10	14	900	10	0.85
Geometria_32x32	gini	350	70	13	900	10	0.81
Geometria_64x64	gini	350	10	14	900	11	0.81
Atributo_16x16	gini	10	30	12	900	5	0.91
Atributo_32x32	gini	10	70	14	900	11	0.90
Atributo_64x64	gini	10	70	14	900	11	0.89

Esses ajustes de hiperparâmetros foram utilizados para treinar os modelos de aprendizado de máquina finais. Cada modelo foi treinado utilizando todos os dados coletados para o respectivo *dataset*. O treinamento foi realizado com o algoritmo de árvore de decisão da biblioteca Scikit-learn (Pedregosa et al., 2011) e com os hiperparâmetros ajustados. Em seguida, uma nova versão do TMC2 foi criada, baseada na implementação da abordagem apresentada no Capítulo 4 e utilizando os novos modelos treinados. Essa nova versão do TMC2 foi utilizada para os testes experimentais.

5.1.2 Abordagem de Interrupção Antecipada de PUs

A segunda abordagem especializada proposta baseia-se na interrupção antecipada de PUs. Essa abordagem utiliza modelos de aprendizado de máquina treinados com dados extraídos do processo de codificação do V-PCC para tomar decisões quanto à escolha dos modos de particionamento de PUs. Para isso, um *dataset* foi criado e modelos de árvores de decisão foram treinados. Esta seção apresenta a metodologia utilizada pela abordagem proposta, bem como os resultados experimentais obtidos com sua aplicação.

Cada CU folha pode ainda ser dividida em duas ou quatro PUs, que são preditas de forma independente. Todas as PUs de uma CU são preditas com predição interquadros ou intraquadro. O melhor tipo de particionamento é escolhido pelo processo de RDO do HM, o qual testa todas as possibilidades disponíveis. A mesma estratégia de interrupção antecipada utilizada na abordagem focada nas CTUs pode ser adaptada para uso nas PUs. Ao encontrar o melhor tipo de particionamento, não é necessário testar os particionamentos restantes, tornando possível reduzir o tempo de codificação necessário.

A estratégia utilizada nesta seção segue aquela apresentada em Correa et al. (2014b), na qual as PUs são testadas com os modos MSM e $2N \times 2N$ e, após isso, modelos de aprendizado de máquina são utilizados para avaliar se os demais modos de particionamento precisam ser considerados. Em caso positivo, os demais modos são avaliados; caso contrário, a busca é interrompida e um dos modos já avaliados

é escolhido para aquela CU. Essa estratégia é aplicada em PUs preditas com predição interquadros, onde a maior quantidade de modos de particionamento é avaliada, sendo utilizada em todas as profundidades da CTU, ou seja, 64×64 , 32×32 , 16×16 e 8×8 . Os dados apresentados em Correa et al. (2014b) indicam que a grande maioria das PUs é codificada sem ser subdividida em PUs menores, ou seja, utilizando os modos MSM e $2N \times 2N$. Por este motivo, esses modos foram escolhidos para serem sempre testados.

Semelhantemente à abordagem de interrupção de CTUs, caso o melhor modo de particionamento seja MSM ou $2N \times 2N$ e a busca seja interrompida, não haverá perdas de eficiência de codificação, pois o melhor modo está entre os avaliados, mas haverá ganhos de tempo de codificação. Caso a busca não seja interrompida e o melhor particionamento seja MSE ou $2N \times 2N$, não haverá ganho de tempo de codificação, pois o melhor modo de particionamento está entre os avaliados e será escolhido pelo processo de RDO.

Para os casos em que o melhor modo de particionamento está entre os demais modos, ou seja, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$ ou $nR \times 2N$, não haverá ganhos de tempo de codificação, pois, nesses casos, todos os particionamentos devem ser avaliados para encontrar o particionamento de PU ótimo. Caso a busca seja interrompida antecipadamente, haverá perda de eficiência de codificação, pois o melhor modo de particionamento não está entre os modos avaliados.

Para treinar os modelos de aprendizado de máquina, foram extraídos dados de codificações dos subfluxos de geometria e atributos do V-PCC. Foram utilizadas as mesmas sequências de teste citadas na Seção 5.2, com precisão de 11 *bits*. As *features* coletadas são listadas a seguir:

- **Ratio($2N \times 2N$, MSM):** Razão entre os custos de RD dos modos de PU $2N \times 2N$ e MSM.
- **NormDiff($2N \times 2N$, MSM):** Diferença normalizada entre os custos de RD dos modos $2N \times 2N$ e MSM. Calculada conforme a Equação 6.
- **RD(MSM), RD($2N \times 2N$) e RD(best):** Custos de RD obtidos para os modos MSM, $2N \times 2N$ e para o melhor modo avaliado até o momento, respectivamente.
- **UpperCU_div:** Indica se a CU superior foi subdividida ou não.
- **Ratio(best, MSM):** Razão entre o custo de RD do melhor modo avaliado até o momento e o custo de RD do modo MSM.
- **NormDiff(best, MSM):** Diferença normalizada entre o custo de RD do melhor modo avaliado até o momento e o custo de RD do modo MSM. Calculada conforme a Equação 6.

Adicionalmente, uma *feature* indicando qual particionamento de PU foi selecionado para cada CU foi coletada para ser usada como valor-alvo. Se o modo de particionamento selecionado é MSM ou $2N \times 2N$, esse valor é definido como 1, caso contrário, é definido como 0. Essas *features* foram coletadas utilizando a versão modificada do TMC2 para realizar a coleta das informações sobre PUs. Os dados coletados foram organizados em oito *datasets* diferentes, de acordo com o subfluxo e profundidade de CTU: Geometria 64x64, Geometria 32x32, Geometria 16x16, Geometria 8x8, Atributos 64x64, Atributos 32x32, Atributos 16x16 e Atributos 8x8.

Uma etapa de balanceamento de dados foi realizada em cada *dataset*, considerando o número de exemplos da classe minoritária para cada sequência e taxa. O menor valor da classe minoritária foi utilizado para subamostrar os exemplos das demais classes e taxas, igualando assim o número de exemplos das classes 1 e 0.

O algoritmo de árvore de decisão foi utilizado para treinar os modelos de aprendizado de máquina. Uma busca de hiperparâmetros foi realizada utilizando o espaço de busca apresentado na Tabela 9, com o algoritmo de busca aleatória (*RandomizedSearchCV*) e o número de iterações fixado em 1820 (1% do total de combinações). Foi utilizada validação cruzada com cinco *folds* e a métrica *F1-score* para avaliação dos modelos treinados.

Tabela 9 – Espaço de busca de hiperparâmetros para o ajuste dos modelos de árvores de decisão.

Hiperparâmetro	Espaço de Busca	Tamanho
a. Criterion	{'gini', 'entropy'}	2
b. min_samples_split	[10–90 (passo 20)] \cup [100–1150 (passo 250)]	10
c. min_samples_leaf	[10–90 (passo 20)] \cup [100–1150 (passo 250)]	10
d. max_depth	[2–14 (passo 1)]	13
e. max_leaf_nodes	[10–90 (passo 20)] \cup [100–900 (passo 200)]	10
f. max_features	[1–7 (passo 1)]	7
Total		182.000

Os resultados da busca aleatória para ajuste dos hiperparâmetros dos modelos de árvores de decisão treinados para as PUs estão apresentados na Tabela 10. Observa-se que os valores de F1-Score variam conforme o subfluxo e a profundidade da CTU, sendo mais elevados para CUs menores (como 8x8 e 16x16) e para o subfluxo de atributos. Por exemplo, para os *datasets* Atributo_8x8 e Geometria_8x8, os modelos alcançaram F1-Scores de 0,99, indicando excelente capacidade de discriminação entre as classes. Para CUs maiores, como 64x64, os F1-Scores foram menores, especialmente para Geometria_64x64 (0,57), o que pode ser atribuído à menor quantidade de exemplos ou à maior dificuldade do modelo em capturar padrões nesses tamanhos. Esses ajustes de hiperparâmetros foram utilizados para treinar os modelos empregados nos experimentos de avaliação da abordagem proposta. Os modelos finais foram treinados utilizando todo o *dataset* de treino e incorporados em uma nova versão do

TMC2 22.1, modificada para interrupção antecipada do processo de avaliação das PUs.

Tabela 10 – Resultados da busca aleatória para cada *dataset*.

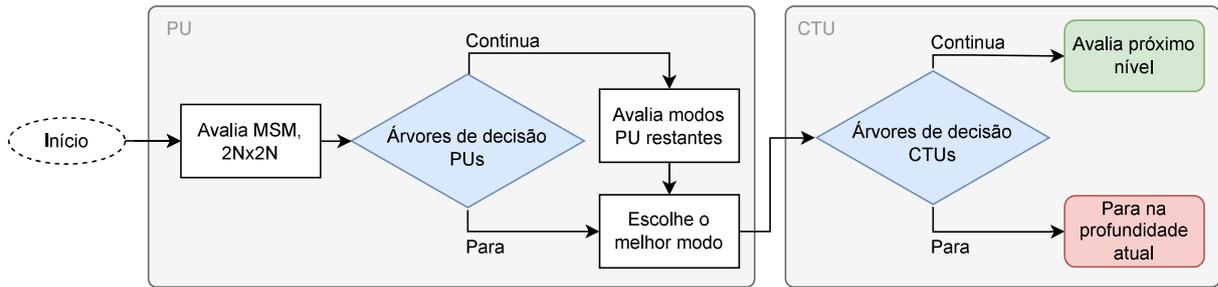
Dataset	Hiperparâmetro (conforme Tabela 9)						F1-Score
	a	b	c	d	e	f	
Atributo_8x8	entropy	1100	30	7	10	6	0.99
Atributo_16x16	entropy	100	10	10	700	1	0.90
Atributo_32x32	gini	1100	350	14	700	3	0.84
Atributo_64x64	entropy	600	350	13	900	7	0.76
Geometria_8x8	gini	30	30	13	30	5	0.99
Geometria_16x16	gini	70	70	13	700	1	0.78
Geometria_32x32	gini	850	30	13	900	1	0.70
Geometria_64x64	entropy	350	600	14	900	1	0.57

5.1.3 Solução Combinada de Interrupção Antecipada de CTUs e PUs

A solução combinada integra as duas abordagens de interrupção antecipada, utilizando os modelos de aprendizado de máquina treinados para CTUs e PUs. Para isso, para cada CU avaliada, inicialmente são testados os modos MSM e $2N \times 2N$, conforme a abordagem da Seção 5.3. Em seguida, as árvores de decisão treinadas para PUs são utilizadas para decidir se os demais modos de particionamento devem ser avaliados ou não. Se a decisão for positiva, os demais modos são avaliados e o melhor modo é escolhido pelo processo de RDO. Caso contrário, a busca é interrompida e o melhor modo entre MSM e $2N \times 2N$ é selecionado. Após essa etapa, aplica-se a abordagem de interrupção antecipada de CTUs. Ou seja, após a avaliação dos modos de particionamento de PUs, o modelo treinado para CTUs é utilizado para decidir se a CU deve ser subdividida ou não. Se a decisão for positiva, a CU é subdividida e as CUs do próximo nível da CTU são avaliadas. Caso contrário, o processo de codificação daquela CU é interrompido e o processo de RDO escolhe o melhor modo de particionamento entre os avaliados. A Figura 18 apresenta um fluxograma da solução combinada, ilustrando o processo de funcionamento da abordagem proposta para CTUs e PUs.

Esse processo é realizado nas profundidades de CTU de 64x64, 32x32 e 16x16 para a interrupção antecipada de CTUs, e nas profundidades de 64x64, 32x32, 16x16 e 8x8 para a interrupção antecipada de PUs. A solução combinada foi implementada em uma nova versão do TMC2 22.1, modificada para incorporar as duas soluções especializadas, aplicando-as nos subfluxos de geometria e atributos, conforme descrito nas seções anteriores. Espera-se que essa abordagem integrada proporcione uma redução ainda maior no tempo de codificação; no entanto, o impacto tende a ser menor do que a soma dos impactos das soluções isoladas, pois ambas atuam em níveis diferentes do processo de codificação. Além disso, a interrupção antecipada de CTUs

Figura 18 – Fluxograma da solução combinada.



Fonte: Autor.

pode reduzir o número de PUs avaliadas, tornando a interrupção antecipada de PUs menos impactante.

5.2 Resultados para Interrupção Antecipada de CTUs

A fim de avaliar o desempenho da abordagem proposta para interrupção antecipada de CTUs, foram realizados testes utilizando as sequências de teste indicadas na CTC (MPEG, 2020) do V-PCC. Foram utilizadas as sequências de teste de 10 *bits* *longdress*, *loot*, *queen*, *redandblack*, *soldier*, e também as sequências de teste com 11 *bits* de precisão, *dancer* e *basketball_player*. Embora os modelos tenham sido treinados com dados de 11 *bits*, as sequências de teste de 10 *bits* foram incluídas para avaliar o desempenho da abordagem em diferentes níveis de precisão.

As sete sequências de teste foram codificadas utilizando a configuração temporal RA em cada uma das cinco configurações de taxa. Para cada sequência, foram utilizadas as primeiras 64 imagens. Os experimentos foram realizados tanto com o *software* de referência TMC2 original quanto com a versão modificada que incorpora o método proposto.

Para cada experimento, foram calculados os valores de PSNR para a geometria das nuvens de pontos reconstruídas, utilizando as métricas ponto-a-ponto (D1) e ponto-a-plano (D2). O PSNR para a informação de atributos foi derivado dos canais de Luminância (Luma), Crominância Azul (Cb) e Crominância Vermelha (Cr). Essas métricas foram computadas com a ferramenta *Distortion Metric*, conforme especificado na CTC. Além disso, foram registrados os tempos de codificação dos subfluxos de vídeo para avaliar o desempenho da codificação. Os experimentos foram realizados utilizando a versão 22.1 do TMC2 (MPEG, 2024) em um sistema com processador Intel Xeon E5-4650 e 512 GB de RAM.

Os resultados de redução de tempo de codificação obtidos nos experimentos realizados são apresentados na Tabela 11. Os resultados mostram que a abordagem pro-

posta conseguiu reduzir o tempo de codificação em média 35,06% para o tempo total, 28,23% para o subfluxo de geometria e 43,18% para o subfluxo de atributos. Os valores de redução de tempo de codificação variaram entre as sequências de teste, com a sequência *queen* apresentando a maior redução (45,73% no tempo total, 40,84% no subfluxo de geometria e 51,00% no subfluxo de atributos) e a sequência *dancer* apresentando a menor redução (28,21% no tempo total, 22,24% no subfluxo de geometria e 35,05% no subfluxo de atributos). Quando comparada com a abordagem adaptada de vídeos 2D, que obteve uma redução média de 48,19% no tempo total, 42,96% para geometria e 54,72% para atributos, a abordagem especializada apresentou uma redução de tempo menor. No entanto, essa diferença pode ser justificada pelo fato de os modelos treinados com dados do V-PCC serem mais adequados ao contexto das nuvens de pontos dinâmicas, resultando em decisões mais precisas e menos agressivas.

Tabela 11 – Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de CTUs.

Sequência de Teste	Red. Tempo		
	Total	Geometria	Atributo
longdress	-29,10%	-25,13%	-33,86%
loot	-37,64%	-25,39%	-53,34%
queen	-45,73%	-40,84%	-51,00%
redandblack	-30,11%	-23,82%	-37,61%
soldier	-43,04%	-35,31%	-52,11%
basketball	-31,59%	-24,90%	-39,27%
dancer	-28,21%	-22,24%	-35,05%
Média	-35,06%	-28,23%	-43,18%

Os resultados de BD-Rate obtidos para os subfluxos de atributos são apresentados na Tabela 12. Os resultados médios de BD-Rate foram de 0,49% para Luma, -0,53% para Cb e -0,06% para Cr, considerando o Attr.BD-AttrRate. Observando os resultados obtidos para cada sequência de teste, a sequência *queen* apresentou o maior aumento de BD-Rate para Luma (1,71%), enquanto as demais sequências apresentaram valores próximos ou abaixo da média. Esse valor discrepante, novamente, pode ser justificado pelo fato de a sequência *queen* ser a única criada a partir de conteúdo sintético, o que pode dificultar a generalização dos modelos treinados predominantemente com dados de cenas reais. Para os canais de cor Cb e Cr, os resultados médios indicam um pequeno aumento na eficiência de codificação, chegando a quase 2% para o canal Cb da sequência *dancer*. Esses resultados mostram que a abordagem não prejudicou a qualidade da nuvem de pontos reconstruída e também não prejudicou a taxa de *bits* do conteúdo comprimido.

Além disso, os resultados para o *Attr.BD-TotalRate* também foram próximos dos valores observados para o Attr.BD-AttrRate, indicando que o impacto da abordagem

nos atributos se mantém mesmo quando se considera o *bitrate* total do *bitstream*. Vale lembrar que o Attr.BD-TotalRate leva em conta não apenas o subfluxo de atributos, mas também o subfluxo de geometria, refletindo o impacto global da abordagem sobre a eficiência de compressão do V-PCC. Como discutido no Capítulo 4, os ganhos de eficiência podem ser originados de vários fatores, como o uso de vários *layers*, CTUs em uma região sem pontos válidos, entre outros.

Comparando com a abordagem adaptada de vídeos 2D, observa-se que os valores de BD-Rate para a abordagem especializada foram significativamente inferiores. Enquanto a abordagem adaptada apresentou um aumento médio de 3,89% em Attr.BD-AttrRate para Luma e 6,25% em Attr.BD-TotalRate, a abordagem especializada manteve os valores próximos de zero, indicando um impacto praticamente nulo na eficiência de codificação dos atributos. Isso evidencia que o uso de modelos treinados especificamente para o contexto do V-PCC resulta em decisões mais adequadas, evitando perdas de eficiência observadas na abordagem adaptada.

Tabela 12 – Resultados de BD-Rate de Atributos para a abordagem especializada de interrupção antecipada de CTUs.

Sequência de Teste	Attr.BD-AttrRate			Attr.BD-TotalRate		
	Luma	Cb	Cr	Luma	Cb	Cr
longdress	0,24%	-0,05%	0,25%	0,22%	-0,05%	0,17%
loot	0,12%	0,98%	-0,96%	-0,02%	0,54%	-0,69%
queen	1,71%	-1,36%	0,71%	1,47%	-0,57%	0,80%
redandblack	0,34%	-0,12%	0,02%	0,40%	0,08%	0,13%
soldier	0,13%	-1,35%	-1,20%	-0,01%	-0,59%	-0,93%
basketball	0,50%	0,51%	1,36%	0,50%	0,12%	0,92%
dancer	0,37%	-1,82%	-0,61%	0,41%	-1,01%	-0,32%
Média	0,49%	-0,53%	-0,06%	0,42%	-0,25%	0,01%

Os resultados de BD-Rate relacionados aos subfluxos de geometria são apresentados na Tabela 13. Aqui são observadas as maiores diferenças entre a abordagem especializada e a abordagem adaptada de vídeos 2D. Para a abordagem adaptada, os valores médios de BD-Rate foram de 4,23% para D1 e 4,63% para D2 considerando o Geom.BD-GeomRate, indicando uma perda significativa de eficiência de codificação nos subfluxos de geometria. Já para a abordagem especializada, os valores médios de BD-Rate foram próximos de zero (respectivamente -0,07% para D1 e 0,00% para D2), mostrando que praticamente não houve impacto negativo na eficiência de codificação da geometria. Isso evidencia que os modelos treinados especificamente para o contexto do V-PCC conseguem tomar decisões mais adequadas para os subfluxos de geometria, evitando as perdas observadas com a abordagem adaptada de vídeos 2D. Além disso, ao analisar o Geom.BD-TotalRate, observa-se o mesmo comportamento: enquanto a abordagem adaptada apresentou valores médios de 4,83% (D1) e 4,99% (D2), a abordagem especializada manteve os valores próximos de zero. Esses resul-

tados reforçam a importância de utilizar modelos especializados para o domínio das nuvens de pontos dinâmicas, principalmente para preservar a eficiência de compressão dos subfluxos de geometria.

Tabela 13 – Resultados de BD-Rate de Geometria para a abordagem especializada de interrupção antecipada de CTUs.

Sequência de Teste	Geom.BD-GeomRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
longdress	-0,03%	0,07%	-0,01%	0,05%
loot	-0,26%	-0,31%	-0,15%	-0,21%
queen	-0,04%	0,13%	-0,39%	-0,31%
redandblack	0,01%	0,21%	-0,10%	0,05%
soldier	-0,54%	-0,44%	-0,29%	-0,28%
basketball	0,24%	0,13%	0,09%	-0,02%
dancer	0,16%	0,20%	-0,01%	0,02%
Média	-0,07%	0,00%	-0,12%	-0,10%

A Tabela 14 traz a relação de BD-Rate e redução de tempo de codificação obtidas pela abordagem especializada. Os resultados médios obtidos mostram que, para as métricas Geom.BD-GeomRate e Attr.BD-AttrRate, a abordagem especializada conseguiu manter os valores próximos de zero, indicando que não houve impacto negativo significativo na eficiência de codificação. A média de BD-Rate para D1 foi de -0,002%, para D2 foi de 0,001%, enquanto para Luma foi de 0,011%, para Cb foi de -0,013% e para Cr foi de 0%.

Ao comparar com a abordagem adaptada de vídeos 2D apresentada no Capítulo 4, observa-se que os valores médios dessa razão para a abordagem adaptada foram consideravelmente maiores, principalmente para os subfluxos de geometria, onde foram obtidos os valores de 0,105 para D1 e 0,120 para D2. Isso confirma a hipótese de que os modelos especializados apresentam resultados superiores a modelos adaptados de vídeos 2D, quando considerado o contexto de codificação de subfluxos de geometria. Comparando os valores relacionados aos subfluxos de atributos, a abordagem adaptada de vídeos 2D apresentou valores médios de 0,059 para Luma, 0,008 para Cb e 0,032 para Cr. A diferença entre os valores obtidos pela abordagem especializada e a abordagem adaptada de vídeos 2D para atributos não é tão grande quando comparada com os subfluxos de geometria, mas, ainda assim, houve uma melhora do *trade-off* entre eficiência de codificação e redução de tempo de codificação em todos os valores observados. Isso indica que soluções feitas para vídeos 2D podem ser adaptadas para o contexto de compressão de subfluxos de atributos no V-PCC, sem comprometer significativamente a eficiência de codificação, mas que soluções especializadas conseguem alcançar resultados ainda melhores.

Os resultados obtidos pela abordagem de interrupção antecipada de CTUs especializada mostram que a abordagem conseguiu uma melhor relação entre eficiência

Tabela 14 – Razão de BD-Rate e redução de tempo de codificação para a abordagem especializada de interrupção antecipada de CTUs.

Sequência	BD-Rate / Red. Tempo				
	Geom.BD-GeomRate		Attr.BD-AttrRate		
	D1	D2	Luma	Cb	Cr
longdress	-0,001	0,003	0,007	-0,002	0,007
loot	-0,010	-0,012	0,018	0,002	-0,018
queen	-0,001	0,003	0,033	-0,027	0,014
redandblack	0,000	0,009	0,009	-0,003	0,000
soldier	-0,015	-0,013	0,004	-0,020	-0,023
basketball	0,010	0,005	0,018	0,005	0,005
dancer	0,007	0,009	0,011	-0,052	-0,017
Média	-0,002	0,001	0,011	-0,013	0,000

de codificação e redução de tempo de codificação, quando comparada com a abordagem adaptada de vídeos 2D. O uso de modelos de aprendizado de máquina treinados especificamente para o contexto do V-PCC permitiu decisões mais precisas e menos agressivas, resultando em uma redução significativa no tempo de codificação sem comprometer a eficiência de compressão dos subfluxos de atributos e geometria. Dessa forma, a abordagem proposta representa um avanço significativo para a viabilização do V-PCC em cenários que possuam recursos limitados, sem comprometer a eficiência de codificação.

5.3 Resultados para Interrupção Antecipada de PUs

A fim de avaliar a abordagem de interrupção antecipada das PUs do V-PCC proposta nesta seção, novos experimentos foram realizados. Nesses experimentos, os primeiros 64 quadros de cada sequência de teste de 10 *bits* e 11 *bits* indicadas nas CTCs do V-PCC (MPEG, 2020) foram codificados, considerando as cinco configurações de taxa e a configuração temporal RA. Esses experimentos foram realizados em um computador com processador Intel Xeon E5-4650 e 512 GB de RAM.

Os resultados de qualidade objetiva de geometria (D1 e D2) e atributos (PSNR dos canais Luma, Cb e Cr) das nuvens de pontos reconstruídas foram calculados com a ferramenta *Distortion Metric* e comparados com os resultados obtidos com a implementação padrão do V-PCC. Além disso, os resultados de taxa de *bits* e tempo de codificação de cada experimento também foram coletados.

A Tabela 15 apresenta os resultados de redução de tempo de codificação dos subfluxos de geometria, atributos e tempo total do HEVC. Os resultados mostram que a abordagem proposta alcançou uma redução de tempo de codificação média de 45,68% para subfluxos de geometria e 49,75% para subfluxos de atributo, com 47,54% de redução no tempo total do HEVC. Esses resultados mostram que a abordagem con-

segiu uma boa redução do tempo de codificação, reduzindo o tempo necessário para codificar os subfluxos para quase metade do tempo original. As maiores reduções totais foram obtidas nas sequências *queen* (com 50,66%) e *soldier* (com 50,38%). A sequência *dancer* obteve os menores valores de redução de tempo de codificação total e também nos subfluxos de geometria e atributo.

Tabela 15 – Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de PUs.

Sequência de Teste	Red. Tempo		
	Total	Geometria	Atributo
longdress	-46,49%	-44,78%	-48,79%
loot	-48,57%	-45,85%	-52,33%
queen	-50,66%	-49,47%	-52,07%
redandblack	-46,71%	-44,73%	-49,29%
soldier	-50,38%	-48,22%	-53,03%
basketball	-45,08%	-43,56%	-46,94%
dancer	-44,29%	-43,13%	-45,79%
Média	-47,45%	-45,68%	-49,75%

Os resultados de BD-Rate para os atributos, considerando as taxas de *bits* de cada subfluxo e total, são apresentados na Tabela 16. Os resultados médios mostram que o impacto na eficiência de codificação, considerando Attr.BD-AttrRate, foi de 1,69% para o canal Luma e 2,4% e 2,44% para os canais Cb e Cr, respectivamente. O impacto na eficiência de codificação foi maior nos canais de crominância, chegando a 5,09% para o canal Cr da sequência *basketball* e 3,22% para o canal Cb das sequências *loot* e *longdress*. No canal Cr da sequência *loot* foi observado um ganho de eficiência de codificação de 1,53%.

Esses resultados indicam que, embora a abordagem proposta tenha proporcionado uma redução significativa no tempo de codificação, houve um aumento moderado no BD-Rate dos atributos, principalmente nos canais de crominância. Esse aumento pode ser atribuído ao fato de que, ao interromper antecipadamente a avaliação dos modos de particionamento das PUs, algumas decisões subótimas podem ser tomadas pelo modelo de aprendizado de máquina, resultando em uma leve perda de eficiência de compressão dos atributos. No entanto, cabe destacar que os valores médios de BD-Rate permanecem em níveis considerados aceitáveis, principalmente quando comparados ao expressivo ganho de tempo obtido.

Além disso, é possível que o impacto no canal Luma, que é o mais relevante para a percepção visual, tenha sido inferior a 2% em média, enquanto os maiores aumentos ocorreram nos canais Cb e Cr, que possuem menor impacto na percepção visual. Assim, o impacto perceptível na qualidade visual dos atributos pode ser menor do que os valores de BD-Rate sugerem.

A Tabela 17 mostra os resultados de BD-Rate para a geometria. Os resultados

Tabela 16 – Resultados de Attr.BD-AttrRate e Attr.BD-TotalRate para a abordagem especializada de interrupção antecipada de PUs.

Sequência de Teste	Attr.BD-AttrRate			Attr.BD-TotalRate		
	Luma	Cb	Cr	Luma	Cb	Cr
longdress	2,68%	3,22%	3,30%	1,86%	2,32%	2,37%
loot	2,06%	3,22%	-1,53%	1,01%	1,53%	-1,03%
queen	1,19%	3,19%	2,75%	0,80%	2,17%	1,86%
redandblack	2,01%	2,37%	1,70%	1,18%	1,44%	0,98%
soldier	1,24%	1,89%	2,52%	0,96%	1,61%	1,92%
basketball	1,29%	1,44%	5,09%	0,79%	0,97%	3,17%
dancer	1,35%	1,49%	1,87%	0,80%	1,01%	1,20%
Média	1,69%	2,40%	2,44%	1,06%	1,58%	1,50%

Geom.BD-GeomRate médios foram de -0,25% para D1 e -0,26% para D2, o que indica que o uso da interrupção antecipada das PUs não afetou a eficiência de codificação de forma negativa. Dentre as sequências de teste utilizadas, apenas a métrica D2 da sequência *soldier* apresentou um valor positivo de BD-Rate (0,09%), sendo que todas as demais apresentaram um leve aumento de eficiência de codificação em ambas as métricas. O ganho de eficiência não foi observado em Geom.BD-TotalRate, o que era o esperado, pois essa métrica também considera a taxa de *bits* dos atributos, os quais apresentaram uma leve piora na eficiência de codificação.

Tabela 17 – Resultados de Geom.BD-GeomRate e Geom.BD-TotalRate para a abordagem especializada de interrupção antecipada de PUs.

Sequência de Teste	Geom.BD-GeomRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
longdress	-0,23%	-0,30%	0,85%	0,71%
loot	-0,42%	-0,42%	0,37%	0,36%
queen	-0,38%	-0,16%	0,59%	0,73%
redandblack	-0,27%	-0,35%	0,72%	0,61%
soldier	-0,09%	0,09%	0,93%	0,99%
basketball	-0,16%	-0,32%	0,34%	0,12%
dancer	-0,21%	-0,38%	0,34%	0,09%
Média	-0,25%	-0,26%	0,59%	0,52%

A relação de BD-Rate por redução de tempo de codificação é apresentada na Tabela 18. Os resultados mostram que, especialmente para geometria, a abordagem conseguiu reduzir o tempo de codificação de forma eficiente, obtendo valores de -0,005 para D1 e -0,006 para D2. Para os atributos, os valores foram maiores, com 0,032 para Luma, 0,048 para Cb e 0,046 para Cr. Esses valores contrastam com os resultados de F1-score obtidos para cada modelo na Tabela 9, que mostravam resultados melhores para atributos. Isso indica que, para os subfluxos de geometria, PUs interrompidas antecipadamente de forma incorreta não estão afetando a qualidade final da geometria na nuvem de pontos reconstruída. Ou seja, mantendo a mesma taxa

de *bits* no subfluxo de geometria, perdas de qualidade em certos blocos não estão afetando a qualidade da nuvem de pontos final. O mesmo não é observado nos atributos, onde o aumento no BD-Rate indica que a interrupção antecipada das PUs pode impactar negativamente a eficiência de compressão dos canais de cor, resultando em uma leve perda de qualidade nos atributos da nuvem de pontos reconstruída.

Tabela 18 – Razão de BD-Rate e redução de tempo de codificação para a abordagem especializada de interrupção antecipada de PUs.

Sequência	BD-Rate / Red. Tempo				
	Geom.BD-GeomRate		Attr.BD-AttrRate		
	D1	D2	Luma	Cb	Cr
longdress	-0,005	-0,007	0,055	0,066	0,068
loot	-0,009	-0,009	0,039	0,061	-0,029
queen	-0,008	-0,003	0,023	0,061	0,053
redandblack	-0,006	-0,008	0,041	0,048	0,034
soldier	-0,002	0,002	0,023	0,036	0,048
basketball	-0,004	-0,007	0,027	0,031	0,104
dancer	-0,005	-0,009	0,029	0,033	0,041
Média	-0,005	-0,006	0,034	0,048	0,046

Comparando com os resultados da Seção 5.2, é possível notar que os resultados para geometria são superiores, apresentando uma melhor relação de perda de eficiência de codificação por ganho de tempo. Para os atributos, embora o aumento no BD-Rate seja mais perceptível, os valores ainda permanecem em níveis considerados aceitáveis, especialmente diante da redução do tempo de codificação obtida.

5.4 Resultados para Interrupção Antecipada de CTUs e PUs

Experimentos foram realizados para avaliar a solução combinada de interrupção antecipada de CTUs e PUs. Foram utilizadas as mesmas sequências de teste e configurações de taxa dos experimentos anteriores, codificando os primeiros 64 quadros de cada sequência de teste de 10 *bits* e 11 *bits* do V-PCC, com a configuração temporal RA e as cinco configurações de taxa de *bits*. Os resultados obtidos foram coletados e comparados com aqueles da implementação padrão do V-PCC, conforme descrito nas seções anteriores.

Os resultados de tempo de codificação da solução combinada são apresentados na Tabela 19. Observa-se que a solução combinada alcançou uma redução média de 60,05% no tempo de codificação total, com reduções de 57,79% para o subfluxo de geometria e 63,21% para o subfluxo de atributos. Todas as sequências apresentaram redução de tempo de codificação total superior a 55%, destacando-se as sequências *loot* e *soldier*, que atingiram reduções de 66,63% e 63,04%, respectivamente. Para os atributos, a redução de tempo de codificação foi ainda mais expressiva, chegando

a 68,44% na sequência *loot*. Esses resultados indicam que a solução combinada proporcionou uma redução significativa no tempo de codificação e manteve consistência nos resultados, demonstrando que, mesmo em sequências com características distintas, a solução foi capaz de reduzir o tempo de codificação de forma eficiente.

Tabela 19 – Redução de tempo de codificação dos subfluxos de geometria, atributos e total do HEVC para a solução combinada de interrupção antecipada de CTUs e PUs.

Sequência de Teste	Red. Tempo		
	Total	Geometria	Atributo
longdress	-55,24%	-54,43%	-56,86%
loot	-61,85%	-56,82%	-68,44%
queen	-66,63%	-66,88%	-66,75%
redandblack	-56,50%	-53,92%	-60,01%
soldier	-63,04%	-60,69%	-66,14%
basketball	-59,37%	-56,56%	-62,90%
dancer	-57,75%	-55,25%	-60,98%
Média	-60,05%	-57,79%	-63,21%

Os resultados de BD-Rate para os atributos, considerando as taxas de *bits* de cada subfluxo e total, são apresentados na Tabela 20. Os resultados médios mostram que o impacto na eficiência de codificação, considerando Attr.BD-AttrRate, foi de 1,75% para o canal Luma e de 2,17% e 3,26% para os canais Cb e Cr, respectivamente. Observa-se uma leve piora na eficiência de codificação dos atributos, especialmente nos canais de crominância, com destaque para o canal Cr da sequência *basketball*, que apresentou um aumento de 5,22%. No entanto, os valores médios ainda permanecem em níveis considerados aceitáveis, especialmente diante da significativa redução do tempo de codificação obtida.

Os resultados de Attr.BD-TotalRate foram levemente melhores, com valores de 1,54% para o canal Luma e 1,80% e 2,55% para os canais Cb e Cr, respectivamente. Resultados de BD-TotalRate menores que os de Attr.BD-AttrRate indicam que o restante do processo de codificação, incluindo a geometria, conseguiu compensar a leve piora na eficiência de codificação dos atributos.

A Tabela 21 mostra os resultados de BD-Rate para a geometria. A solução combinada resultou em um aumento médio de 1,16% (D1) e 1,31% (D2) no Geom.BD-GeomRate, indicando uma pequena perda de eficiência de compressão na geometria dos pontos. Considerando o BD-Rate total (Geom.BD-TotalRate), os aumentos médios foram de 1,67% (D1) e 1,78% (D2). Esses resultados são levemente piores do que os obtidos com as soluções isoladas de interrupção antecipada de CTUs e PUs. No entanto, houve um aumento significativo na redução do tempo de codificação, o que pode compensar a leve piora na eficiência de compressão da geometria, principalmente em situações em que o tempo de codificação é um fator crítico, como em aplicações em tempo real ou em cenários com recursos computacionais limitados.

Tabela 20 – Resultados de Attr.BD-AttrRate e Attr.BD-TotalRate para a solução combinada de interrupção antecipada de CTUs e PUs.

Sequência de Teste	Attr.BD-AttrRate			Attr.BD-TotalRate		
	Luma	Cb	Cr	Luma	Cb	Cr
longdress	2,65%	3,20%	3,57%	2,03%	2,47%	2,74%
loot	1,83%	3,16%	2,94%	1,48%	2,11%	2,00%
queen	1,86%	3,41%	2,35%	1,53%	2,33%	1,97%
redandblack	2,05%	2,36%	1,71%	1,55%	1,77%	1,31%
soldier	1,38%	0,67%	3,23%	1,50%	1,05%	2,82%
basketball	1,12%	1,27%	5,22%	1,31%	1,49%	4,02%
dancer	1,36%	1,14%	3,81%	1,39%	1,38%	2,97%
Média	1,75%	2,17%	3,26%	1,54%	1,80%	2,55%

Tabela 21 – Resultados de Geom.BD-GeomRate e Geom.BD-TotalRate para a solução combinada de interrupção antecipada de CTUs e PUs.

Sequência de Teste	Geom.BD-GeomRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
longdress	0,90%	1,01%	1,86%	2,04%
loot	1,03%	0,94%	1,48%	1,27%
queen	1,36%	1,91%	2,41%	2,93%
redandblack	0,77%	0,72%	1,40%	1,32%
soldier	1,53%	2,00%	2,20%	2,67%
basketball	1,39%	1,35%	1,32%	1,16%
dancer	1,11%	1,21%	1,05%	1,08%
Média	1,16%	1,31%	1,67%	1,78%

A relação de BD-Rate por redução de tempo de codificação é apresentada na Tabela 22, considerando Geom.BD-GeomRate e Attr.BD-AttrRate. Os valores obtidos para subfluxos de geometria mostram uma melhor relação de BD-Rate por redução de tempo de codificação, com médias de 0,02 para D1 e 0,022 para D2. Para os subfluxos de atributo, os valores foram um pouco maiores: 0,028 para Luma, 0,035 para Cb e 0,052 para Cr. Os resultados para o canal Cr indicam uma relação pior em comparação com os demais, sugerindo que a interrupção antecipada de CTUs e PUs pode ter um impacto maior na eficiência de compressão dos canais de crominância.

No geral, os valores obtidos apresentam uma boa relação de BD-Rate por redução de tempo, mantendo-se em níveis aceitáveis, especialmente considerando a significativa redução do tempo de codificação. Esses resultados indicam que a solução combinada proporcionou uma redução expressiva no tempo de codificação, com impacto moderado na eficiência de compressão dos subfluxos de geometria e atributos.

A Tabela 23 apresenta uma comparação entre as soluções propostas, considerando a redução de tempo de codificação, o impacto em BD-Rate e a relação BD-Rate por redução de tempo. Considerando a redução de tempo de codificação, a abordagem combinada (CTU+PU) apresentou uma redução total média de 60,05%, superando as demais soluções. Quando analisado o impacto em BD-Rate por redução de

Tabela 22 – Razão de BD-Rate e redução de tempo de codificação para a solução combinada de interrupção antecipada de CTUs e PUs.

Sequência	BD-Rate / Red. Tempo				
	Geom.BD-GeomRate		Attr.BD-AttrRate		
	D1	D2	Luma	Cb	Cr
longdress	0,017	0,019	0,047	0,056	0,063
loot	0,018	0,017	0,027	0,046	0,043
queen	0,020	0,029	0,028	0,051	0,035
redandblack	0,014	0,013	0,034	0,039	0,028
soldier	0,025	0,033	0,021	0,010	0,049
basketball	0,025	0,024	0,018	0,020	0,083
dancer	0,020	0,022	0,022	0,019	0,062
Média	0,020	0,022	0,028	0,035	0,052

tempo, a abordagem que apresentou os melhores resultados foi a especializada de interrupção antecipada de CTUs (CTU), com relação BD-Rate por redução de tempo de 0,011 para D1 e 0,001 para D2. Considerando apenas o subfluxo de geometria, a abordagem especializada de interrupção antecipada de PUs (PU) apresentou os melhores resultados, com relação de -0,006 para D2, indicando um leve aumento na eficiência de compressão da geometria. A abordagem adaptada do contexto de vídeos 2D (CTU 2D) obteve bons resultados de redução de tempo de codificação, mas apresentou um aumento significativo no BD-Rate. O uso da abordagem PU apresenta resultados de redução de tempo similares, porém com impacto menor no BD-Rate, especialmente para o subfluxo de geometria. A abordagem combinada (CTU+PU) também apresentou uma relação BD-Rate melhor que a abordagem CTU 2D, além de maior redução de tempo de codificação. Ambas as abordagens individuais (CTU e PU) apresentaram bons resultados de BD-Rate para geometria (0% e -0,26%, respectivamente). No entanto, ao combinar as abordagens, nota um aumento na perda de eficiência de codificação (1,31%). Esse resultado vai contra o esperado, ao combinar duas abordagens com bons resultados, era esperado também obter bons resultados. Isso mostra que, ao combinar duas estratégias que individualmente apresentam bom desempenho, pode haver uma sobreposição de decisões subótimas ou um acúmulo de erros que resulta em maior perda de eficiência de compressão do que o observado nas abordagens isoladas. Esse efeito pode estar relacionado ao fato de que a aplicação simultânea das duas técnicas reduz ainda mais o número de particionamentos avaliados, aumentando a probabilidade de escolhas subótimas pelo processo de RDO.

As três abordagens especializadas (CTU, PU e CTU+PU) apresentaram uma redução de tempo de codificação significativa, sendo que o impacto em BD-Rate aumenta conforme a redução de custo computacional da abordagem aumenta. Assim, essas soluções podem ser utilizadas de forma complementar, dependendo do cenário e dos

requisitos de codificação. A abordagem CTU+PU, por exemplo, pode ser empregada em situações em que a redução de tempo de codificação é mais crítica, enquanto a abordagem CTU pode ser escolhida quando a eficiência de compressão é mais importante, mas ainda se deseja reduzir o tempo de codificação. A abordagem PU representa um meio-termo entre as duas, sendo uma boa opção para cenários em que se busca uma redução significativa do tempo de codificação, mas ainda se deseja manter uma eficiência de compressão razoável. Embora a relação de BD-Rate e tempo para atributos seja inferior para a abordagem PU, quando comparada com CTU+PU, os resultados para geometria são superiores. Se o foco for a geometria, a abordagem PU pode ser uma boa escolha, pois apresenta uma redução de tempo de codificação significativa e uma leve melhora na eficiência de compressão da geometria. Também é possível utilizar soluções diferentes para cada subfluxo, como, por exemplo, adotar a abordagem CTU para o subfluxo de atributo e a abordagem PU para o subfluxo de geometria, que apresentaram os melhores resultados de BD-Rate/Tempo para cada subfluxo correspondente.

Tabela 23 – Comparação das abordagens propostas quanto à redução de tempo, impacto em BD-Rate e relação BD-Rate/Red. Tempo.

Solução	Red. Tempo			BD-Rate		BD-Rate/Tempo	
	Total	Atr	Geo	Atr Luma	Geo D2	Atr Luma	Geo D2
CTU 2D	-48,19%	-54,72%	-42,96%	3,89%	4,63%	0,059	0,120
CTU	-35,06%	-43,18%	-28,23%	0,49%	0,00%	0,011	0,001
PU	-47,45%	-49,75%	-45,68%	1,69%	-0,26%	0,034	-0,006
CTU+PU	-60,05%	-63,21%	-57,79%	1,75%	1,31%	0,028	0,022

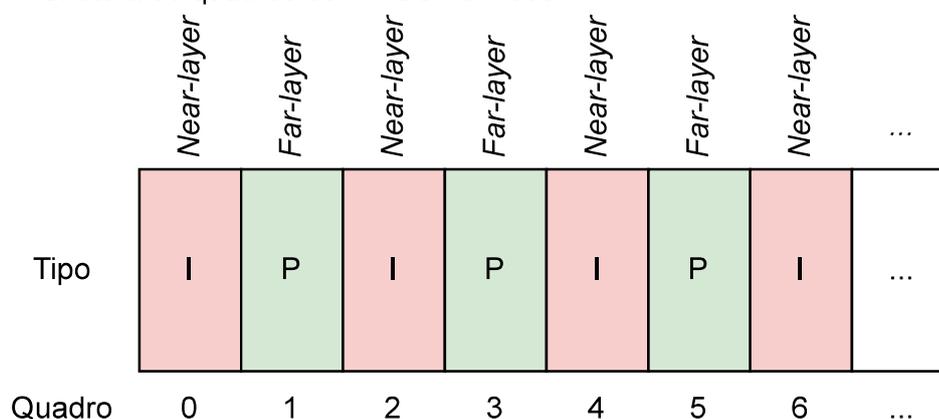
5.4.1 Resultados para o Modo AI

As soluções propostas nas seções anteriores foram desenvolvidas para funcionar apenas em blocos com predição inter do HEVC e foram avaliadas com o modo RA do V-PCC. No entanto, o V-PCC também possui um modo de codificação AI, que é utilizado pela maioria dos trabalhos encontrados na literatura sobre redução do custo computacional do V-PCC.

Apesar do nome, o modo AI do V-PCC não implica que todos os blocos dos subfluxos de vídeo sejam codificados exclusivamente com predição intra pelo HEVC. No contexto do V-PCC, o termo All Intra refere-se principalmente à organização temporal dos *patches*, ou seja, não há preocupação em manter *patches* correspondentes em posições próximas entre quadros vizinhos. Ao avaliar a estrutura de quadros dos subfluxos de vídeo gerados pelo V-PCC ao utilizar o modo AI, observou-se a presença de quadros do tipo I, que são codificados exclusivamente com predição intra, e quadros do tipo P, cujos blocos são codificados com predição inter ou intra. Mais especificamente, os quadros do tipo I são usados para codificar quadros de *near-layer*, enquanto

os quadros do tipo P são usados para codificar quadros de *far-layer*. A estrutura de quadros do V-PCC no modo AI é composta por uma sequência de quadros do tipo I e P, como mostrado na Figura 19. No modo RA, apenas o primeiro quadro é do tipo I, enquanto os demais quadros são do tipo P.

Figura 19 – Estrutura de quadros do V-PCC no modo AI.



Fonte: Autor.

Diante disso, as soluções propostas nas seções anteriores podem ser aplicadas ao modo AI do V-PCC. No entanto, é importante ressaltar que as soluções apresentadas neste trabalho são aplicadas apenas em blocos preditos com modo inter do HEVC, e sua efetividade está restrita aos casos em que há predição interquadros nos subfluxos.

A fim de permitir uma comparação com trabalhos relacionados avaliados no modo AI do V-PCC, foi realizado um experimento utilizando a solução combinada de interrupção antecipada de CTUs e PUs. Esses experimentos seguiram a mesma metodologia e as mesmas sequências utilizadas nos experimentos das seções anteriores.

Os resultados de redução de tempo de codificação desses experimentos são apresentados na Tabela 24. Os resultados mostram que a solução combinada, usando o modo AI do V-PCC, alcançou uma redução média de 42,73% no tempo de codificação total, com uma redução de 46,99% para o subfluxo de geometria e 37,88% para o subfluxo de atributos. Os valores de redução de tempo são inferiores aos obtidos com o modo RA, o que já era esperado, pois a solução é aplicada em metade dos quadros no modo AI, ou seja, apenas nos quadros do tipo P. Nesses quadros, a predição intra ainda pode ser utilizada, limitando a efetividade da solução. Mesmo assim, a diferença de redução de tempo de codificação, quando comparada ao modo RA, é de 28,84% (60,05% contra 42,73%, considerando o tempo total), ou seja, mesmo sendo aplicada em metade dos quadros, a solução ainda conseguiu mais da metade da redução de tempo de codificação obtida no modo RA. Isso indica que os quadros do tipo P representam uma quantidade significativa do tempo total de codificação no modo AI e que a solução proposta ainda é efetiva nesse modo do V-PCC.

Tabela 24 – Resultados de tempo de codificação da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.

Sequência de Teste	Red. Tempo		
	Total	Geometria	Atributo
longdress	-39,76%	-43,98%	-35,74%
loot	-44,52%	-48,47%	-39,76%
queen	-46,03%	-53,02%	-37,84%
redandblack	-41,66%	-44,87%	-38,02%
soldier	-41,70%	-47,56%	-35,12%
basketball	-42,68%	-45,16%	-39,74%
dancer	-42,74%	-45,85%	-38,91%
Média	-42,73%	-46,99%	-37,88%

Os resultados de BD-Rate para os atributos e geometria são apresentados nas Tabelas 25 e 26, respectivamente. Os resultados médios de BD-Rate para os atributos mostram um valor médio de 0,02% para o canal Luma, -0,17% para o canal Cb e -0,05% para o canal Cr, considerando Attr.BD-AttrRate. Para geometria, os resultados médios de BD-Rate foram de -0,04% para D1 e 0,05% para D2, considerando Geom.BD-GeomRate. Esses resultados foram superiores aos obtidos com a solução combinada no modo RA, o que era esperado, pois a solução é aplicada em metade dos quadros no modo AI, reduzindo o impacto na eficiência de compressão. Esses resultados mostram que as perdas de qualidade nos quadros *far-layer* não afetam tanto a eficiência de codificação dos subfluxos de vídeo.

Tabela 25 – Resultados de BD-Rate para Atributos da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.

Sequência de Teste	Attr.BD-AttrRate			Attr.BD-TotalRate		
	Luma	Cb	Cr	Luma	Cb	Cr
basketball	0,09%	0,15%	0,12%	0,04%	0,09%	0,07%
dancer	-0,01%	0,26%	-0,60%	-0,06%	0,16%	-0,53%
longdress	0,08%	-0,07%	0,52%	0,00%	-0,09%	0,34%
loot	0,04%	-0,07%	-0,03%	-0,07%	-0,15%	-0,12%
queen	-0,01%	-0,50%	0,06%	-0,08%	-0,46%	-0,08%
redandblack	-0,01%	-0,85%	-0,35%	-0,03%	-0,61%	-0,20%
soldier	-0,04%	-0,12%	-0,06%	-0,04%	-0,09%	-0,07%
Média	0,02%	-0,17%	-0,05%	-0,04%	-0,16%	-0,09%

Os resultados de BD-Rate por redução de tempo de codificação para a solução combinada no modo AI do V-PCC são apresentados na Tabela 27. Os valores obtidos para subfluxos de geometria mostram uma relação média de BD-Rate por redução de tempo de codificação de -0,001 para D1 e 0,001 para D2. Os valores médios para os subfluxos de atributo foram de 0,001 para Luma, -0,005 para Cb e -0,001 para Cr. Esses resultados mostram que, mesmo com as limitações impostas ao usar a solução proposta no modo AI, a relação de BD-Rate por redução de tempo obtida se manteve

Tabela 26 – Resultados de BD-Rate para Geometria da solução combinada de interrupção antecipada de CTUs e PUs para o modo AI do V-PCC.

Sequência de Teste	Geom.BD-GeomRate		Geom.BD-TotalRate	
	D1	D2	D1	D2
basketball	-0,13%	0,04%	0,19%	0,48%
dancer	0,16%	0,12%	0,44%	0,41%
longdress	-0,29%	-0,23%	0,00%	0,19%
loot	-0,11%	-0,13%	0,22%	0,27%
queen	-0,06%	0,10%	0,23%	0,51%
redandblack	0,08%	0,14%	0,19%	0,20%
soldier	0,07%	0,28%	0,18%	0,41%
Média	-0,04%	0,05%	0,21%	0,35%

em bons níveis.

Tabela 27 – Razão de BD-Rate e redução de tempo de codificação para a solução combinada de interrupção antecipada de CTUs e PUs no modo AI do V-PCC.

Sequência	BD-Rate / Red. Tempo				
	Geom.BD-GeomRate		Attr.BD-AttrRate		
	D1	D2	Luma	Cb	Cr
longdress	-0,003	0,001	0,003	0,004	0,003
loot	0,003	0,002	0,000	0,007	-0,015
queen	-0,005	-0,004	0,002	-0,002	0,014
redandblack	-0,002	-0,003	0,001	-0,002	-0,001
soldier	-0,001	0,002	0,000	-0,014	0,002
basketball	0,002	0,003	0,000	-0,021	-0,009
dancer	0,002	0,006	-0,001	-0,003	-0,002
Média	-0,001	0,001	0,001	-0,005	-0,001

Ao utilizar a solução combinada no modo AI do V-PCC, foi possível obter bons resultados de redução de custo computacional na etapa de codificação dos subfluxos do V-PCC. Os impactos na eficiência de codificação observados são desprezíveis, com ganhos de eficiência observados principalmente nos canais de crominância e na geometria. No entanto, cabe destacar que esses ganhos de eficiência de codificação não são resultados diretos da aplicação da solução de interrupção antecipada de CTUs e PUs, uma vez que a solução apenas interrompe o processo de RDO quando é avaliado que um bom bloco foi encontrado. Isso, por si só, não gera ganhos de eficiência de codificação; pelo contrário, caso os modelos façam uma escolha equivocada, um bloco subótimo pode ser selecionado, o que leva a uma piora na eficiência de codificação.

Esses ganhos são resultado da forma como a qualidade da nuvem de pontos é avaliada, sendo considerada apenas após a reconstrução. Assim, perdas de qualidade objetiva que são visíveis nos subfluxos após a codificação podem não se propagar para a nuvem de pontos final. Por exemplo, perdas de qualidade em regiões entre os *patches* não afetam a nuvem reconstruída. As características das nuvens de

pontos utilizadas no CTC também podem influenciar esses resultados. Essas nuvens são pré-processadas, passando por etapas para "limpar" a nuvem de pontos, removendo pontos sobrepostos e regiões descontínuas. Isso afeta diretamente o uso do *far-layer*, pois esses quadros mapeiam uma quantidade muito pequena de pontos da nuvem original, influenciando minimamente a nuvem de pontos reconstruída. Dessa forma, os ganhos de eficiência observados nos experimentos não são garantidos em outras sequências de nuvens de pontos dinâmicas com características diferentes das sequências do CTC do V-PCC, sendo um efeito colateral indireto do uso da solução.

5.4.2 Comparações com Trabalhos Relacionados

A Tabela 28 apresenta uma comparação dos resultados médios obtidos com a solução combinada (CTU+PU) e trabalhos relacionados encontrados na literatura focados na redução do tempo da etapa de codificação de vídeo do V-PCC. Para o modo RA, a solução proposta obteve a maior redução do custo computacional total, quando comparada com Lin et al. (2021) e Que; Li (2023). Cabe destacar que apenas Lin et al. (2021) utilizou todas as sequências indicadas no CTC do V-PCC (10 e 11 *bits*), enquanto Que; Li (2023) testou utilizando apenas as sequências de 10 *bits*. Comparando os resultados de BD-Rate, é possível observar que CTU+PU obteve as maiores perdas de eficiência de codificação, com resultados mais próximos aos trabalhos relacionados para os valores de geometria. Os valores da razão de BD-Rate por redução de tempo também mostram uma melhor relação para os trabalhos relacionados, porém, cabe destacar que os valores de BD-Rate por redução de tempo são aproximados, já que os trabalhos relacionados não indicam qual a redução de tempo para cada subfluxo.

Comparando a solução proposta com trabalhos focados no modo AI do V-PCC, a solução combinada não obteve os maiores resultados de redução de tempo, o que já era esperado, pois ela é aplicada apenas nos quadros P. No entanto, a solução obteve bons resultados de BD-Rate, principalmente para geometria, onde apresentou os melhores resultados de BD-Rate e na relação de BD-Rate por redução de tempo. Cabe destacar que nenhum dos trabalhos relacionados utilizou todas as sequências de teste do CTC do V-PCC, empregando apenas sequências de 10 *bits* de precisão, sendo que Gao et al. (2023) ainda utiliza apenas um subconjunto das sequências de 10 *bits*. Song; Liu; Zhang (2025) não indica qual configuração temporal utiliza, mas, por comparar com trabalhos focados no modo AI, assume-se que este também utilize o modo AI.

No geral, a solução obteve resultados competitivos, comparáveis aos obtidos em trabalhos relacionados. Para o modo RA, a solução obteve a maior redução de tempo de codificação. Já para o modo AI, a solução apresentou os melhores resultados de BD-Rate ao considerar os resultados de atributo e geometria, e também obteve

Tabela 28 – Comparação das soluções propostas com trabalhos relacionados quanto à redução de tempo, impacto em BD-Rate e relação BD-Rate por redução de Tempo.

Modo	Solução	Versão TMC2	Red. Tempo			BD-Rate		BD-Rate/Tempo	
			Total	Atr	Geo	Atr Luma	Geo D2	Atr Luma	Geo D2
RA	CTU+PU (RA)	22.1	-60,05%	-63,21%	-57,79%	1,75%	1,31%	0,028	0,022
	Lin et al. (2021)	8.0	-55,43%	–	–	-1,15%	0,82%	-0,020*	0,014*
	Que; Li (2023)	18.0	-43,13%	–	–	0,10%	0,40%	0,002*	0,009*
AI	CTU+PU (AI)	22.1	-42,73%	-46,99%	-37,88%	0,02%	0,05%	0,001	0,001
	Gao et al. (2023)	18.0	-57,80%	-58,51%	-58,25%	0,14%	0,30%	0,002	0,005
	Li et al. (2024)	18.0	–	–	-69,3%	–	0,50%	–	0,007
	Song et al. (2025)	18.0	-54,75%	-57,37	-54,43	0,12%	0,56%	0,002	0,010

*Valores aproximados, pois os trabalhos não apresentam a redução de tempo para cada subfluxo separadamente.

a melhor relação de BD-Rate por redução de tempo. Cabe destacar que todos os trabalhos relacionados foram implementados em versões mais antigas do TMC2 e a maioria não avaliou utilizando todas as sete sequências indicadas nas CTCs do V-PCC; apenas Lin et al. (2021) utilizou todas as sequências, mas se trata de um trabalho antigo, utilizando a versão 8 do TMC2. Assim, nenhuma comparação apresentada é totalmente justa, devido às diferentes condições de avaliação e à ausência de alguns resultados, como a redução de tempo de cada subfluxo.

6 CONCLUSÃO

Essa dissertação apresentou soluções para reduzir o custo computacional da etapa de codificação de vídeos do *software* de referência do V-PCC, com foco na redução do tempo de codificação dos subfluxos de vídeo. Foram propostas quatro soluções baseadas no uso de modelos de aprendizado de máquina, sendo uma adaptada do contexto de vídeos 2D convencionais e três especializadas para o contexto de codificação de nuvens de pontos dinâmicas no V-PCC.

Uma análise da complexidade do V-PCC indicou que a etapa de codificação de vídeo é a mais complexa, com os subfluxos de geometria e atributos representando a maior parte do tempo total de codificação. Um esquema de redução do custo computacional baseado na interrupção antecipada de CTUs do contexto de vídeos 2D comuns teve seu uso analisado no V-PCC. Essa abordagem foi implementada no TMC2 e avaliada com as sequências do CTC do V-PCC, obtendo uma redução média de 48,19% no tempo de codificação total, com impactos significativos de BD-Rate, especialmente para o subfluxo de geometria, onde foram obtidos valores próximos a 5% de aumento no BD-Rate. Esses resultados indicam que, embora essa abordagem possa reduzir o tempo de codificação, ela não é adequada para o contexto do V-PCC, pois os impactos na eficiência de compressão são significativos.

Após isso, foram propostas três abordagens especializadas para o contexto do V-PCC, utilizando modelos de aprendizado de máquina treinados com dados extraídos do próprio processo de codificação. Os modelos foram treinados a partir desses dados, uma etapa de balanceamento foi aplicada para evitar viés de classes, um ajuste de hiperparâmetros foi realizado para otimizar o desempenho dos modelos, e eles foram incorporados ao funcionamento do *software* de referência do V-PCC.

A primeira abordagem foi uma adaptação da abordagem do contexto de vídeos 2D, com novos modelos de aprendizado de máquina treinados com dados do V-PCC, que obteve uma redução média de 35,06%, com impactos de BD-Rate de 0,49% para o subfluxo de atributos (Luma) e -0,07% para o subfluxo de geometria (D1). A relação de BD-Rate por redução de tempo foi superior àquela obtida com a abordagem adaptada do contexto de vídeos 2D. Esses resultados indicaram que soluções especializadas

para o contexto do V-PCC podem reduzir o tempo de codificação com impactos menores na eficiência de compressão. A segunda abordagem especializada foi baseada na interrupção antecipada de PUs, que obteve uma redução média de 47,45% no tempo de codificação total, com impactos de BD-Rate de 1,69% para o subfluxo de atributos (Luma) e -0,26% para o subfluxo de geometria (D2). A terceira abordagem especializada combinou as duas soluções anteriores, utilizando interrupção antecipada de CTUs e PUs, e obteve uma redução média de 60,05% no tempo de codificação total, com impactos de BD-Rate de 1,75% para o subfluxo de atributos (Luma) e 1,31% para o subfluxo de geometria (D2).

A solução integrada também foi avaliada no modo AI do V-PCC, obtendo uma redução média de 37,17% no tempo de codificação total, com impactos de BD-Rate de 0,40% para o subfluxo de atributos (Luma) e 0,12% para o subfluxo de geometria (D2). Esses resultados mostraram que a solução proposta é efetiva tanto no modo RA quanto no modo AI do V-PCC. Comparações com trabalhos da literatura mostraram que as soluções propostas são competitivas ou superiores, especialmente no modo RA, em que a solução CTU+PU obteve a maior redução de tempo de codificação, com impactos controlados na eficiência de compressão. No modo AI, a solução também apresentou bons resultados, especialmente para o subfluxo de geometria, onde obteve os melhores resultados de BD-Rate e na relação de BD-Rate por redução de tempo.

Como trabalhos futuros, pretende-se investigar a aplicação de outras técnicas de aprendizado de máquina, como redes neurais profundas, para melhorar ainda mais a eficiência das soluções propostas. Além disso, ainda é possível ajustar os modelos de aprendizado de máquina para favorecer a escolha de decisões que não impactem na eficiência de compressão, aumentando o peso das classes que não levam à interrupção antecipada de CTUs e PUs. Isso levaria a uma menor redução de tempo de codificação, mas com impactos ainda menores na eficiência de compressão. Também espera-se uma exploração de soluções para redução do custo computacional de codificação dos blocos intra, que potencializarão os ganhos de tempo obtidos no modo AI do V-PCC, principalmente, e também pode gerar ganhos de tempo do modo RA, visto que blocos intra também são usados nesse modo.

REFERÊNCIAS

- ANG, P. H.; RUETZ, P. A.; AULD, D. Video compression makes big gains. **IEEE Spectrum**, [S.l.], v.28, n.10, p.16–19, 1991.
- BJONTEGAARD, G. Calculation of average PSNR differences between RD-curves. **VCEG-M33**, [S.l.], 2001.
- BROSS, B.; WANG, Y.-K.; YE, Y.; LIU, S.; CHEN, J.; SULLIVAN, G. J.; OHM, J.-R. Overview of the Versatile Video Coding (VVC) standard and its applications. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, n.10, p.3736–3764, 2021.
- BURKOV, A. **The hundred-page machine learning book**. [S.l.]: Andriy Burkov Quebec City, QC, Canada, 2019. v.1.
- CAO, C. **3D point cloud compression**. 2021. Tese (Doutorado em Ciência da Computação) — Institut Polytechnique de Paris.
- CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, [S.l.], v.16, p.321–357, 2002.
- CHO, S.; KIM, M. Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.23, n.9, p.1555–1564, 2013.
- COLLIS, R. Lidar. **Applied optics**, [S.l.], v.9, n.8, p.1782–1788, 1970.
- CORREA, G.; ASSUNCAO, P. A.; AGOSTINI, L. V.; SILVA CRUZ, L. A. da. Fast HEVC encoding decisions using data mining. **IEEE transactions on circuits and systems for video technology**, [S.l.], v.25, n.4, p.660–673, 2014.
- CORREA, G.; ASSUNCAO, P.; SILVA CRUZ, L. A. da; AGOSTINI, L. Classification-based early termination for coding tree structure decision in HEVC. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS (ICECS), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.239–242.

FENLASON, J. **Gprof**. 2024. Accessed: 2024-04-15, Disponível em: https://ftp.gnu.org/old-gnu/Manuals/gprof-2.9.1/html_mono/gprof.html.

GAO, W.; YUAN, H.; LI, G.; LI, Z.; YUAN, H. Low Complexity Coding Unit Decision for Video-Based Point Cloud Compression. **IEEE Transactions on Image Processing**, [S.l.], v.33, p.149–162, 2023.

GAUTIER, G.; MERCAT, A.; FRÉNEAU, L.; PITKÄNEN, M.; VANNE, J. UVG-VPC: Voxelized Point Cloud Dataset for Visual Volumetric Video-based Coding. In: INTERNATIONAL CONFERENCE ON QUALITY OF MULTIMEDIA EXPERIENCE (QOMEX), 2023., 2023. **Anais...** [S.l.: s.n.], 2023. p.244–247.

GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagem. **Pearson, ISBN-10: 8576054019**, [S.l.], v.10, p.11–27, 2010.

GRAZIOSI, D.; NAKAGAMI, O.; KUMA, S.; ZAGHETTO, A.; SUZUKI, T.; TABATABAI, A. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). **APSIPA Transactions on Signal and Information Processing**, [S.l.], v.9, p.e13, 2020.

GROIS, D.; NGUYEN, T.; MARPE, D. Coding efficiency comparison of av1/vp9, h. 265/mpeg-hevc, and h. 264/mpeg-avc encoders. In: PICTURE CODING SYMPOSIUM (PCS), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.1–5.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), 2008., 2008. **Anais...** [S.l.: s.n.], 2008. p.1322–1328.

International Organization for Standardization. **ISO/IEC 23090-5:2021 – Information technology – Coded representation of immersive media – Part 5: Video-based point cloud compression**. 2021. Standard published by ISO/IEC JTC 1/SC 29, Disponível em: <https://www.iso.org/standard/79331.html>.

JIA, Q.; KIM, K.; LEE, M. K.; JANG, E. S. Simple grid-based refinement segmentation algorithm for MPEG video-based point cloud compression. **IEEE Access**, [S.l.], v.12, p.23695–23706, 2024.

KIM, I.-K.; MIN, J.; LEE, T.; HAN, W.-J.; PARK, J. Block partitioning structure in the HEVC standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.22, n.12, p.1697–1706, 2012.

KIM, J.; KIM, Y.-H. Fast grid-based refining segmentation method in video-based point cloud compression. **IEEE Access**, [S.l.], v.9, p.80088–80099, 2021.

LI, G.; GAO, W.; GAO, W. **Point Cloud Compression: Technologies and Standardization**. [S.l.]: Springer Nature, 2024.

LI, T.; XU, M.; DENG, X. A deep convolutional neural network approach for complexity reduction on intra-mode HEVC. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1255–1260.

LI, Y.; HUANG, J.; WANG, C.; HUANG, H. Unsupervised learning-based fast CU size decision for geometry videos in V-PCC. **Journal of Real-Time Image Processing**, [S.l.], v.21, n.1, p.11, 2024.

LIN, T.-L.; BU, H.-B.; CHEN, Y.-C.; YANG, J.-R.; LIANG, C.-F.; JIANG, K.-H.; LIN, C.-H.; YUE, X.-F. Efficient quadtree search for HEVC coding units for V-PCC. **IEEE Access**, [S.l.], v.9, p.139109–139121, 2021.

MPEG. Common Test Conditions for V3C and V-PCC. **ISO/IEC JTC 1/SC 29/WG 11**, [S.l.], 2020.

MPEG. **HEVC Test Model Version 16.20 Screen Content Model Version 8.8**. 2021. 2021. Disponível em: <https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20+SCM-8.8/>.

MPEG. **Video Point Cloud Compression - VPCC - mpeg-pcc-tmc2 test model candidate software**. 2024. Disponível em: <<https://github.com/MPEGGroup/mpeg-pcc-tmc2>>.

PARMAR, A.; KATARIYA, R.; PATEL, V. A review on random forest: An ensemble classifier. In: INTERNATIONAL CONFERENCE ON INTELLIGENT DATA COMMUNICATION TECHNOLOGIES AND INTERNET OF THINGS, 2018. **Anais...** [S.l.: s.n.], 2018. p.758–763.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISSEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in Python. **the Journal of machine Learning research**, [S.l.], v.12, p.2825–2830, 2011.

PREDA, M. **V-PCC codec description**. 2020. Disponível em: <https://www.mpeg.org/wp-content/uploads/mpeg_meetings/134_OnLine/w20352.zip>.

QUE, S.; LI, Y. Portable Perceptron Network-Based Fast Mode Decision for Video-Based Point Cloud Compression. **CAAI Artificial Intelligence Research**, [S.l.], v.2, 2023.

RASCHKA, S.; MIRJALILI, V. **Python machine learning**: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. [S.l.]: Packt Publishing Ltd, 2019.

REHBEIN, G.; COSTA, E.; CORRÊA, G.; SANTOS, C.; PORTO, M. A Machine-Learning-Driven Fast Video-based Point Cloud Compression (V-PCC). In: BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB, 30., 2024, Porto Alegre, RS, Brasil. **Proceedings...** SBC, 2024. p.20–27.

SALDANHA, M.; SANCHEZ, G.; MARCON, C.; AGOSTINI, L. Learning-Based Complexity Reduction Scheme for VVC Intra-Frame Prediction. In: INTERNATIONAL CONFERENCE ON VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p.1–5.

SANTOS, C.; REHBEIN, G.; COSTA, E.; CORRÊA, G.; PORTO, M. Efficiency and complexity analysis of video-based and geometry-based point cloud encoders. **Journal of Real-Time Image Processing**, [S.l.], v.22, n.3, p.114, May 2025.

SCHAEFER, R. Call for proposals for point cloud compression V2. In: ISO/IEC JTC1 SC29/WG11 MPEG, 117TH MEETING. HOBART, TAS, 2017. **Anais...** [S.l.: s.n.], 2017.

SCHWARZ, S.; PREDA, M.; BARONCINI, V.; BUDAGAVI, M.; CESAR, P.; CHOU, P. A.; COHEN, R. A.; KRIVOKUĆA, M.; LASSERRE, S.; LI, Z. et al. Emerging MPEG standards for point cloud compression. **IEEE Journal on Emerging and Selected Topics in Circuits and Systems**, [S.l.], v.9, n.1, p.133–148, 2018.

SEIDEL, I.; FREITAS, D. R.; DOREA, C.; GARCIA, D. C.; FERREIRA, R. U.; HIGA, R.; QUEIROZ, R. L. de; TESTONI, V. Memory-friendly segmentation refinement for video-based point cloud compression. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p.3383–3387.

SHI, Y.-Q.; SUN, H. **Image and video compression for multimedia engineering**: Fundamentals, algorithms, and standards. [S.l.]: CRC press, 2000.

SONG, W.; LIU, X.; ZHANG, Q. Fast Coding Unit Partitioning Method for Video-Based Point Cloud Compression: Combining Convolutional Neural Networks and Bayesian Optimization. **Electronics**, [S.l.], v.14, n.7, p.1295, 2025.

SULLIVAN, G. J.; OHM, J.-R.; HAN, W.-J.; WIEGAND, T. Overview of the high efficiency video coding (HEVC) standard. **IEEE Transactions on circuits and systems for video technology**, [S.l.], v.22, n.12, p.1649–1668, 2012.

SULLIVAN, G. J.; OHM, J.-R.; HAN, W.-J.; WIEGAND, T. Overview of the High Efficiency Video Coding (HEVC) Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.22, n.12, p.1649–1668, 2012.

SUTHAHARAN, S. Decision tree learning. **Machine learning models and algorithms for big data classification: thinking with examples for effective learning**, [S.l.], p.237–269, 2016.

SZE, V.; BUDAGAVI, M.; SULLIVAN, G. J. High efficiency video coding (HEVC). **Integrated circuit and systems, algorithms and architectures**, [S.l.], v.39, p.40, 2014.

TAYLOR, P. **Global**: fastest fixed internet speed by country 2024 | Statista — statista.com. 2024. Disponível em: <https://www.statista.com/statistics/896772/countries-fastest-average-fixed-broadband-internet-speeds/>.

WANG, Y.; WANG, Y.; CUI, T.; FANG, Z. Fast Video-Based Point Cloud Compression Based on Early Termination and Transformer Model. **IEEE Transactions on Emerging Topics in Computational Intelligence**, [S.l.], 2024.

WIEGAND, T.; SULLIVAN, G. J.; BJONTEGAARD, G.; LUTHRA, A. Overview of the H. 264/AVC video coding standard. **IEEE Transactions on circuits and systems for video technology**, [S.l.], v.13, n.7, p.560–576, 2003.

YANG, H.; SHEN, L.; DONG, X.; DING, Q.; AN, P.; JIANG, G. Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.30, n.6, p.1668–1682, 2020.

YE, J.; CHOW, J.-H.; CHEN, J.; ZHENG, Z. Stochastic gradient boosted distributed decision trees. In: ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 18., 2009. **Proceedings...** [S.l.: s.n.], 2009. p.2061–2064.

Apêndices

APÊNDICE A – Trabalhos Submetidos e Publicados

A seguir são listados os trabalhos submetidos e publicados ao longo do período do mestrado:

A.1 Coding Efficiency and Complexity Analysis of the Geometry-based Point Cloud Encoder

Autores: Cristiano Santos, Leandro Tavares, Eduardo Costa, Gustavo Rehbein, Guilherme Corrêa, Marcelo Porto

Data de publicação: 27/02/2024

Conferência: 2024 IEEE 15th Latin America Symposium on Circuits and Systems (LASCAS)

Abstract: In recent years, 3D point cloud content has gained attention thanks to applications such as virtual, augmented or mixed reality, real-time immersive communications, and autonomous driving systems. However, raw point clouds comprise large amount of data, and compression is mandatory to allow efficient transmission and storage. The MPEG group proposed the standard called Geometry-based Point Cloud Compression (G-PCC) implemented in the Test Condition Category 1 and 3 (TMC13) software. TMC13 can significantly reduce the amount of data in static point clouds, but requires a high computational cost making real-time compression unfeasible, especially for devices with limited computational power and energy resources. This paper presents a computational cost and coding efficiency evaluation of the G-PCC coding tools. Two main contributions are presented: first, a coding efficiency analysis of different G-PCC tools; second, a complexity profiling of the G-PCC coding steps aimed at understanding the computational effort distribution. This work provides the first complexity assessment of G-PCC. The presented results can help in the proposal of complexity reduction approaches for the development of more efficient versions of G-PCC for real-time purposes.

A.2 Adaptive Complexity Control for AV1 Video Encoder Using Machine Learning

Autores: Isis Bender, Gustavo Rehbein, Guilherme Correa, Luciano Agostini, Marcelo Porto

Data de publicação: 19/05/2024

Revista: Journal of Real-Time Image Processing

Abstract: Digital videos are widely used on various platforms, including smartphones and other battery-powered mobile devices, which can suffer from energy consumption and performance constraints. Video encoders are responsible for compressing video data, enabling the use of this type of media by reducing the data rate while maintaining image quality. To promote the use of digital videos, the continuous improvement of digital video encoding standards is crucial. In this context, the Alliance for Open Media (AOM) developed the AV1 (AOMedia Video 1) format. However, the advanced tools and enhancements provided by AV1 come with a high computational cost. To address this issue, this paper presents the learning-based AV1 complexity controller (LACCO). The goal of LACCO is to dynamically optimize the encoding time of the AV1 encoder for HD 1080 and UHD 4K resolution videos. The controller achieves this goal by predicting the encoding time of future frames and classifying input videos according to their characteristics through the use of trained machine learning models. LACCO was integrated into the reference software of the AV1 encoder and its encoding time reduction ranges from 10 to 70%, with average error results ranging from 0.11 to 1.88 percentage points for HD 1080 resolution and from 0.14 to 3.33 percentage points for UHD 4K resolution.

A.3 A Machine-Learning-Driven Fast Video-based Point Cloud Compression (V-PCC)

Autores: Gustavo Rehbein, Eduardo Costa, Guilherme Corrêa, Cristiano Santos, Marcelo Porto

Data de publicação: 14/10/2024

Conferência: Brazilian Symposium on Multimedia and the Web (WebMedia)

Abstract: In recent years, 3D point cloud content has gained attention due to its application possibilities, such as multimedia systems, virtual, augmented, and mixed

reality, through the mapping and visualization of environments and/or 3D objects, real-time immersive communications, and autonomous driving systems. However, raw point clouds demand a large amount of data for their representation, and compression is mandatory to allow efficient transmission and storage. The MPEG group proposed the Video-based Point Cloud Compression (V-PCC) standard, which is a dynamic point cloud encoder based on the use of video encoders through projections into 2D space. However, V-PCC demands a high computational cost, demanding fast implementations for real-time processing and, especially, for mobile device applications. In this paper, a machine-learning-based fast implementation of V-PCC is proposed, where the main approach is the use of trained decision trees to speed up the block partitioning process during the point cloud compression. The results show that the proposed fast V-PCC solution is able to achieve an encoding time reduction of 42.73% for the geometry video sub-stream and 55.3% for the attribute video sub-stream, with a minimal impact on bitrate and objective quality.

A.4 Efficiency and Complexity Analysis of Video-based and Geometry-based Point Cloud Encoders

Autores: Cristiano Santos, Gustavo Rehbein, Eduardo Costa, Guilherme Corrêa, Marcelo Porto

Data de publicação: 19/05/2025

Revista: Journal of Real-Time Image Processing

Abstract: 3D point clouds are increasingly used in applications, such as 3D re-mapping, cultural heritage preservation, and virtual/augmented reality. Given their large data volume, efficient compression is crucial. The MPEG group has introduced two standards: Geometry-based Point Cloud Compression (G-PCC), for static and dynamic-acquired point clouds, and Video-based Point Cloud Compression (V-PCC), specifically for dynamic point clouds. Although both approaches achieve effective data reduction, their high computational complexity limits real-time use, especially on resource-constrained devices. This paper analyzes the computational cost and coding efficiency of G-PCC and V-PCC, identifying the most time-consuming steps. In G-PCC, the Octree-RAHT configuration offers the best trade-off between efficiency and encoding time, with recoloring alone accounting for up to 70%. In V-PCC, video encoding dominates, consuming about 92% of the total time. These findings lay the groundwork for future optimizations to reduce the complexity of more efficient implementations.